

June 1988

Vol. 1 N° 9

Price £1.50

Archive

The Subscription Magazine for Archimedes Users



Watford Video Digitiser Review

Editing the 256-Colour Palette

SigmaSheet & Logistix Reviews

ISO Pascal Library

Screen Conversion Programs

Editing the 256-Colour Palette

OS Commands in BASIC

New Operating Systems for Archimedes



Thanks again, Chaps!

Very many thanks to all those who helped us at the Micro User Show: Ken Biddle, Denis Howlett, Tim Hill, Tudor Jones, Tony Pocock (and son), Matthew Treagus, Chris Walker and David Wild. (I hope I haven't missed anyone out!) Matthew was with us all three days – special thanks to him – but I asked each of the others to do a two or three hour stint on the stand. Most of them didn't seem to want to go when they'd done their bit, so we were well staffed the whole time!

The articles and information keep rolling in – thanks again – it's amazing how just enough arrives each month to fill the pages. I sometimes panic and think there won't be enough, but there always has been so far. And the money too! Just enough of you are buying software and hardware through us to enable us to pay the bills, but, as with the articles, each month there is just enough for our needs. As you must know by now, Sue and I believe that behind it all is One who provides for all our needs and we're grateful to Him as well as to you.

By the way, I understand there's been some discussion about Archive on Prestel. It sounds as if, in some quarters, the technical content goes down well but not the "religious propaganda". Oops, sorry about that! Still, it seems to me that the atmosphere of mutual help and support that surrounds Archive must be, at least partly, due to the large number of committed Christians who are involved, many of whom made themselves known to us at the Show. I believe that this is part of what Jesus talks about as the "Kingdom of God", i.e. that in any group of people where Christian live, work and play, they can and should have a positive influence for good.

I hope I have never been guilty of trying to force my Christian **beliefs** on anyone, but I do not apologise in the slightest for attempting to push Christian **values**. No-one can quarrel with that can they? Anyway, I shall continue to work and pray that "Thy Kingdom come on earth, as it is in heaven".

Paul B.

Archive

Volume 1 • N° 9 • June 1988

Contents

Hardware & Software Available ...2	Contact Box26
Comment Column.....4	SigmaSheet Review28
Readers' Comments5	Logistix – Spreadsheet Plus?31
Hints & Tips6	Using the Debugger – Part 333
Help!!!8	Editing the 256-Colour Palette ...37
Information about Monitors8	Screen Conversions.....41
Matters Arising9	Watford Video Digitiser45
BASIC OVERLAY10	OS Commands in BASIC47
Small Ad's10	More on View50
Sound Synthesis – Part 2.....11	New Operating Systems.....51
FORTTRAN Graphics Library.....22	Colour Printouts52
XED Text Editor Review23	Fact-File53
ISO Pascal24	

Hardware & Software Available

• **Trackerball Adapter.** Pineapple Software have produced an interface to enable owners of the Marconi trackerball to use this device on the Archimedes computer. The adapter has a user port type connector on one end and a 1 m cable with a mouse plug on the other. (£13.95 + VAT)

• **3D CAD / Animation system** from Silicon Vision. They are offering a 3D wire-frame Graphics Development System for £19.95 – “ideal of games and simulations”. (Special price of £17.95 for Archive members – see advert on page 11. Order from Silicon Vision but quote your Archive membership number.) Also under development are, Super-Dump a “near-plotter quality” graphics dump program and 3D Solids Design and Animation system. These are due for release “at the end of June” and users of the simpler system can up-grade for the price difference between the two packages.

• **CADVISE** – Computer Aided Design VISualisation Extended can be used to build up a library of 3D shapes. It comes with standard shapes and these can be ‘welded’ together to make composite shapes. Hidden line removal and perspective views available. £59.95 (or £37.95 for education) from Academic Software.

• **Euclid 3D programmable graphics animation package.** Provides a module which can be called from BASIC or machine code programs to produce orthogonal or perspective drawing with hidden line removal etc etc. £45 from ACE Computing.

• **ArcTFS** – a “free text” database allowing you to cross-reference, file and collate blocks of text of any size – getting away from the fixed data formats of more ‘conventional’ databases. This is a more powerful version of an existing program for the Atari ST. (Reviewed in June edition of ST World) Sounds ideal for anyone trying to handle lots of text – it can even store text in 1st Word Plus format as well as pure

ASCII text and they are hoping to add other WP’s to the list. £29.95 from “Texellence” – should be available by the time you read this.

• **ArcDFS** from Computer Spares Ltd is a(nother) 5.25” disc interface but this one comes with software to read (but not write) DFS discs. We saw it at the M.U.Show and it looks really well made and has a power outlet (as per BBC micro) so that you don’t have to have a disc drive with a built-in power supply unit. The cost is £55.20 for single drive version, £57.50 for the dual drive version but you can subtract £3 if you don’t need the dummy back-plate, i.e. you are going to put it along side an existing podule.

• **Presenter** – a presentation graphics package for £24.95 from Linguinity, the software division of Lindis International Ltd. It uses the WIMP environment to display line, bar and pie charts, in colour or grey-scale (useful for printer-dumps to produce OHP’s). Data can be entered manually or imported from other programs (spreadsheets & databases) and the output can then be used in other packages such as Artisan, Graphic-Writer and 1st Word Plus. (*I saw a development version of Presenter a few weeks ago and it looked very impressive. Ed.*)

• **ServoControl** package from Jansons. Use the printer port to control up to four servos. For £29.95, you get a small hardware interface (you supply the servos), a software module, an interactive instruction manual and a test environment to make sure all the servos are working properly. (See advert on page 21.)

• **ArcImEd** – an art package with a difference, £24.95 from Jansons. If you are writing programs that involves drawing lots of diagrams on the screen, then this program is a must! (*My words, not Jansons – we saw it at the Micro User Show. Ed.*) Draw a picture in the usual mouse-driven, art-package sort of way then at the click of a button, it saves it as a BASIC program! That

program can then be used as a procedure in another program to draw the picture as and when required. It could save you hours in the development of certain types of programs.

- **Modula-2 and FORTH.** Two new language implementations, Modula-2 (£89.95) and RISC FORTH (£69.95) are now available from Blue Grey Software. See the advert on page 27.

- **Expansion Podule** – a single width podule complete with a pre-decoded podule interface providing an area on which small to medium-sized electronics projects can be constructed. This will allow 8-bit memory mapped and/or interrupt driven (IRQ and FIQ) hardware interfaces to be developed very rapidly. Available “late July” as a bare p.c.b. or fully populated with the interface chips. For further details, contact Atomwide Ltd (0689) 38852. (Dealer enquiries only, ring 01-543-2349.)

- **Buffer Podule** – a single width podule allowing you to extend the full bus to devices outside the machine. You can connect simple and external podules via a ribbon cable, (but not MEMC podules because of speed limitations). Still under development but should be ready “in a month or so”. For more details, contact Simon Brise of SGB Computer Services, 140 Disraeli Road, Putney, London, SW15 2DX. (01-874-5675)

- **Archimedes SpellMaster** is here! It checks at over 10,000 words/minute and you can store your user dictionary(ies) in battery-backed ram so that you don't need to load and save them on disc every time you use the computer. The bad news is that on the Acorn ROM board it runs at only about 2,000 words/minute. At the moment it only works on ArcWriter and Inter-Word, but it can be accessed from BASIC and/or ARM code and since it is written as a module, it should not be difficult to link it in with other software.

Compared with BBC Spell-Master, it has extra speed on its *-commands – e.g. the ANAWORD program which I wrote to find as many words as possible from permutations and combinations

of various letters works 2.7 times as fast as on the Master 128.

Extra commands – when you are checking the text, if it finds an unknown word you get five options, three as before: ignore word, correct the word or add it to the dictionary, but you can also now browse the dictionary in order to select the correct word or get it to guess what the word might be.

It also has extra *-commands, so as well as *FUZZY (i.e. give me a word that sounds like...) and *ANAGRAM and *CHECK, there are other commands such as *TYPO which is the equivalent of the guessing facility and *USERTOFILE and *FILETOUSER which can be used for converting ASCII word lists into dictionaries and vice versa.

Free Utilities Disc. If you buy SpellMaster through Archive you get a free disc which contains, as well as a number of utility programs, a ‘word-games’ dictionary – we can't call it a ‘Scrabble’ dictionary but that's what we use it for. One of the utilities allows you to “catch” the words that are output by the various *-commands. (We used it to catch the words given by *CHECK * – i.e. the whole of the dictionary – it produced a file of 580k – all that, as well as the programs, is squeezed into a 128k ROM. Clever, huh?!) Another utility allows you, amongst other things, to do the Daily Express “Target” word game without any thinking! (No, we don't take the Express!!)

Review Software Received...

Apart from reviews already written we have received review copies of the following software: U-Connect Comms Package from Magenta Research Ltd, Flying Start II from Mitre Software, Sign-Writer from Wight Scientific, Euclid (3-D package) from ACE Computing, Architext editor from Hopesoft, Desktop Stories from RESOURCE, CCD's Printer Module, ArcTFS free-form text filing system from Texellence, Logotron Logo and FORTH from BlueGrey Software. **A**

Comment Column

Archimedes Prices going up!

Buy your Archimedes now! The prices are going up! Acorn have announced that the 310's are going up by 8% and the 440's by 10%. UK subscribers can buy their Archimedes through Archive at the old price until the end of June (as long as stocks last) – you will be helping to finance the magazine but you will also get a year's free on-site service and life membership of our Technical Help Service – i.e. free technical telephone calls.

No more free Arc-Writers!

The Arc-Writer offer is at an end, I fear. However, if you already have Arc-Writer, you can send it back to Acorn Direct, Studland Road, Kinsthorpe, Northampton, NN2 6NA, quoting your machine serial number and enclosing a cheque for £45.97 payable to 'Acorn Direct' and they will send you 1st Word Plus (full price £91.94). Beware though that they are saying that the offer is open for one month from the date of the letter (mine was dated 13th May).

If you purchased your machine before 31st March but still have not received Arc-Writer, Acorn should be sending you a half-price voucher for 1st Word Plus. If you've not got your voucher, write to Acorn quoting your machine serial number.

If you purchased your machine after 31st March, there's no freebee, but 1st Word Plus is available through Archive for £85.

Spreadsheet Speed War

In their adverts, Minerva Systems claim that SigmaSheet is the fastest spreadsheet in the world. However, PipeDream now claims to be even faster! Robert Macmillan of Colton Software says that, although speed is not everything, in view of Minerva System's extravagant claims, he thought he ought to point out that the Pipedream spreadsheet, when tested on the PCW benchmarks, is actually 35% faster

than Sigmashet and that this has been verified by Gordon Taylor of A & B Computing.

Hidden Software?

Now there's an idea. You buy a games program and you are told that somewhere, somehow there is a free, hidden space invaders game written entirely in machine code. All you have to do is find it! The authors, Ian and Keith McAlpine of GEM Electronics say that the general home user seems to have lost his will to experiment with software. It is hoped that the incentive of Space Invaders will encourage them to start examining software more closely and thus increase their own proficiency and awareness. (See small ad's for more details.)

Hardware Column

Starting next issue, Brian Cowan will be running a Hardware Column in Archive. Topics covered over the next few months will include an overview of the Archimedes circuits, the RISC chip family set: ARM, MEMC, VIDC and IOC, expanding the memory, disc drives and Winchester, making Podules etc. etc. There will be reviews of commercial hardware products and plenty of hints and tips. Please help by telling us what you would like covered and let us know of any hints or tips you have discovered.

MS-DOS Column

Ken Biddle, who was busy preparing this month's MS-DOS column has landed himself in hospital with a badly torn Achilles tendon! So we hope to have a bumper edition next month. Get well soon, Ken!

Bug or Feature?

Acorn publish a paper called "Archimedes Applications Authors' Guidelines" and in it there is a list of known bugs: 3 in BASIC, 2 in the Debugger, 27 in the MOS, 5 in the Desktop, 11 in Fileswitch and/or ADFS, 1 in Soundchannels, 1 in Podule, 8 in Wimp, 3 in Fonts, 10 in NetFS and 4 in Econet! Some of them are rather obscure but it does strengthen the case of those who reckon that Arthur 2.0 should be made available

for little more than a handling charge rather than being sold as a new piece of software.

New Shareware Disc

We are at present preparing a new shareware disc, so if any of you have programs that you think you would like putting on it, do send them to us in the next week or two. We have already got some excellent graphics animations from Mike (Beard) Williams and we are hoping to put on a high speed version of LIFE and William Doggett's new, improved Mandelbrot program.

Eureka

Eureka! the Archimedes Bulletin Board is now on line – well, sort of! The good news is that we've decided to waive the subscription. Archive subscribers can download any or all of the program from the magazine – the issue 9 programs will be on the board by the time you read this. The bad news is that it isn't working properly yet. There have been some problem with the modem and/or the ACMB software but it is improving, so bear with us. If you have constructive comments about Eureka, do let us know a.s.a.p. Ta! **A**

Readers' Comments

- **Using Z88.** One reader writes that he has had success with a Sinclair Z88 and the Archimedes, linking them via the RS423. He had problems initially, but with the 1.24 version of the serial fix he has successfully transferred files.
- **Unfair to Acorn.** He also reckoned that we were being unfair to Acorn in saying that if the RS423 did not work properly, the Archimedes could be said to be "unfit for the purpose...etc". He continues, "I bought one of the first Archimedes in August 1987 and I felt it was a terrific achievement to get it out even with the bugs it had. I was expecting far more bugs and, overall, I'm very happy with my machine."
- **Life** – Many readers must have been fascinated by Tony Brain's LIFE program. This is the sort of program that benefits from the remarkable speed of the Archimedes. Life

enthusiasts should look at the program in the June issue of A & B Computing. The speed is absolutely phenomenal and the accompanying article explains the algorithm used and how the speed is achieved. (*But wait until you see the Life program on the new Sharware Disc! Ed.*)

• **ROM/RAM podules.** What is the point of this add-on? The paged ROM system on the BBC was conceived to get round the dual problems of lack of memory and slow (tape) storage and it worked well. But the Archimedes comes with a fast disc and plenty of memory as standard, so there is no objection to loading the programs from disc into the RMA when they are required (this only needs to be done once, when the machine is switched on). In fact, I hear a rumour that the software in ROM is downloaded into RMA and hence the podule and ROMs are accessed for only a few milliseconds at most.

Given this, I have concluded that the idea of ROM-based software for the Arc is an invention of Computer Concepts to (a) allow them to charge higher prices for their software on ROM, (b) require people to buy £53+£42 (minimum) worth of extra hardware before being able to use any CC products, (c) make it harder for users to copy (pirate?) their software. I'm not sure what the advantage to the users is – the possibility of running BBC ROMs under the emulator must be regarded as a stopgap similar to harnessing a horse to a Ferrari. Other software houses seem to be distributing similar products on disc at no disadvantage to themselves.

Sorry to have gone on for so long, but I feel that something has to be said to dampen the general enthusiasm regarding the ROM podule. I for one will not be running any CC products on my computer unless they change their distribution policy to one more appropriate to this new machine and I would be greatly heartened if many others were to do likewise.

*N.B. The views expressed here are not necessarily those of the editors of Archive magazine. Let us know what you think. Ed. **A***

Hints and Tips...

• **View B3.0.** In order to get View B3.0 working properly under the 6502 emulator, as well as poking the three bytes at &A8A1 to &EA you have to poke &80C2 - 4 to &EA otherwise, the emulator tries to run it as a second processor ROM image - without any success. With this patch, the address space from &B328 to &BFFF is effectively free for patching. Perhaps some enterprising programmer could use this space for extra routines - e.g. a resident printer-driver.

• **Special characters.** If you want to generate special characters on-screen, try pressing <ctrl-shift-alt> and then one of the ordinary keys and you will find that you can get all sorts of alternative characters such as © and ® and the half, quarter and three-quarter signs (which I haven't even got on the Apple Mac!). What is more, if you are using Arc-Writer, it will actually print some of the symbols, such as the fractions, on the printer!

• **PSU for external 5.25" drives.** If you have an external 5.25" disc drive that does not have its own power supply, you can take the power from the power supply for the hard disc but you will need a special cable and connector. Those who have bought interfaces from Dudley Micro Supplies might be able to get help from them. (Or try the ArcDFS interface mentioned in Hardware & Software Available section.)

• **Aligning decimal points.** If you are trying to print out various figures and want the decimal points to line up above one another, you can use the "fixed format". (See the explanation of the @% variable under the section in the User Guide about the PRINT command.) However, the fixed format prints out trailing zeros, e.g. it prints 234.0000 instead of 234.

The bad news: One reader wanted to avoid this and so was trying to write a BASIC routine to do the aligning and discovered some nasties (presumably due to rounding errors) that gave

the value of LOG100 as 2 but INT(LOG100) as 1! Also he found that LEN(STR\$3.3) was given as 3, but LEN(STR\$4.3) was given as 11!

The good news: You can do a STR\$, use INSTR to find the position of the decimal point and then use PRINT TAB (to position it correctly as in the following example. (Thanks, APL!)

```

10 REM >$ .AlignDP
20
30 REM *****
40 REM * Aligning your decimals *
50 REM * by Adrian Philip Look *
60 REM * 28th April 1988 *
70 REM *****
80
90 REPEAT
100 random=RND(10000)-RND(1)*(RND
                                (1)<.7)
110 PROCalign(20,random)
120 UNTIL FALSE
130 END
140
150 DEFPROCalign(x,number)
                                :REM x = position of DP
160 number$=STR$(number)
170 dot=INSTR(number$,".")
180 IF dot=0 dot=LEN(number$)+1
                                :REM i.e. no DP
190 IF dot=1 dot=0:REM leads with DP
200 PRINT TAB(x-dot);number
                                :REM beware of (x-dot)<0
210 ENDPROC

```

STOP PRESS. The original enquirer about this, Peter Trigg, has just come back with:

```

DEFPROCalign(x%,number)
                                :REM x% = position of DP
LOCAL length%
length%=LEN(STR$(INT(number)))
PRINT TAB(x%-length%);number
ENDPROC

```

This is somewhat neater and it seems to work OK. (This just emphasises the need for sharing ideas - which is why we are setting up our "BASIC utilities" section - see Help!!! section.)

• **Music Editor.** If you have a number of tunes that were prepared using the Music Editor on the 0.2 or 0.3 Welcome discs. You will probably find that all the voices are wrong – usually that the main tune was being played on the percussion! The reason is that the Music Editor refers to the voices by number only, so if the modules are in a different order, the voices are likewise in a different order. It is however possible to change the order of the modules by *RMKILLing the one which is higher in the *ROMMODULES list then doing a *RMTIDY and then *RMREINITing the module which you had just killed.

• **Beware &36D!** Some BBC software (Apollo Mission for example) pokes !877=0 (or !&36D=0). If you transfer this to the Archimedes you will find that the system locks up and neither <ctrl-break> nor <reset> has any effect. Even switching off is ineffective. The only way out is an <R>-power-up. (Do it twice to restore the correct monitor-type setting.)

• **File copying on a single drive.** Are you having difficulty copying files from one disk to another using just one disk drive? Is it that when using the *COPY command with the Prompt option, the system prompts for insertion of the destination disk but never recognises it? If so, issue the command *NODIR first. The system will then read the disk directory every time it attempts to access a disk rather than storing the directory in memory so that disks no longer need *MOUNTing. It is possible to make the change permanent by doing *Configure NODIR.

• **Boot files** that work from desktop or with <shift-break> – set *OPT 4,2 (Run) and create a BASIC program called !BOOT. You can then <shift-break> if configured as language 3 or 4 (desktop or BASIC) or double click the !boot icon from desktop.

If necessary the !BOOT file can contain a line such as: 10*EXEC !BOOT1

!BOOT1 contains tasks such as loading modules that need to be done from supervisor mode.

• **To run Master software,** use *Alphabet Master and *Keyboard Master.

• ***TypeFile again.** After the ideas given last month, I knew there should be an easier way... Clifford Hoggarth has come up with:

```
*Alias$TypeFile Echo ||B|M Type
%0|M Echo ||C|M
```

The point is that on first interpretation, || is turned into | so that it become Echo |B which gives the ASCII 2 (ctrl-B) to turn on the printer and Echo |C to do the reverse.

• **System Delta Plus** – The section in the manual on printer control codes is not very clear. The following points may make it clearer: (1) Clicking on <menu> on the printer control icon brings up a sub-menu giving options for Bold, Compressed or Reset. (2) To select multiple escape sequences, click <select> on the printer control icon, select the Escape icon and enter the first code in the sequence, then press <return>, enter the second code and again press <return>. Repeat this if necessary and click on the OK icon when you have finished. E.g. to get NLQ on an Epson, you want ESC120,1. So use: <select> Escape icon, type 120, press <return>, type 1, press <return>, <select> OK icon.

One limitation of the System Delta Plus package, as it stands, is that you can only create four numeric total fields when producing a "List". The solution, according to Minerva Systems will be to purchase their "Reporter" software – when it becomes available!

• **Taxan Kaga printer** (or the Canon equivalent) – there is a fairly simple way of being able to switch the auto-line feed on and off, to avoid problems with software that may or may not require the printer to auto-line feed.

All you do is to place a switch between two pins of the printer connector. The pins to connect are pin 14 (auto-line feed) and any GND line. When they are connected an auto-line is not performed.

• **Ambiguous *-commands.** If you have two modules loaded into the Archimedes that use the

same *-command name, you can call the command required by: `*<module name>:<command> <....>` So, for example, if you had two modules, 'Utility' and 'Toolkit' both of which have a CATAL command, you can use either `*Utility:CATAL` or `*Toolkit:CATAL` to differentiate between them. **A**

Help!!....

- **"More articles about the basics, please"**, said a number of folk at the Micro User Show. We've had something on using ADFS and are starting assembler for beginners but can anyone say specifically what they want information on and can anyone offer to write any "basic" articles, please? In particular...

- **Article on ARMBE**. We really do need someone to write a very practical guide to using the ARM BASIC Editor. For example, I've only just realised that `<insert>` toggles between insert and over-write mode and so you don't have to go through the preferences menu. I suspect a lot of people are not using ARMBE simply because they are familiar with BASIC's own line editor and haven't got time to learn ARMBE even though it would save them time in the long run. Please write to us if you would like to have a go at this. Thanks.

- **BASIC V utility library**. If you have worked out some neat utilities like the one in Hints & Tips about aligning decimal points, why not send them in to us. Clifford Hoggarth has offered to edit the section, so either send your contributions to us or direct to him at 20 Pinfold Drive, Eccleston Mere, St Helens, WA10 5BT.

- **Education articles**. I'm getting requests for articles relevant to education, but no-one is offering such articles. Can anyone help?!

- **Hard Disc users**. One reader, who has a 440 on order, wants to know if there are any do's and don't's with a hard disc. Are there any hard-disc users who want to comment?

- **View B3.0**. Following on from the hint about View B3.0, John Phelan writes that there is still a problem with the pound key. On the Archimedes, this key returns ASCII 163. Is this redefinable?

- **Calculating π and e**. Brian Cowan asks if anyone knows the way that pi and/or e may be calculated digit by digit on a computer. Presumably they must be all integer algorithms(?)

- **Taxan KP-810 on ArcWriter**. Has anyone worked out how to get a printer driver working for this combination? Drop a line to E. Clinkscales, 68 Wyvis Drive, Nairn, IV12 4TP.

- **Video-titling**. Is there any software, says M Davies of Cardigan, for video-titling? Wild Vision do the hardware for Genlock and Video Overlay, but where is the software?

Also, he asks, is there any software, preferably ROM based, to give NLQ print on his Epson RX80 and also desktop publishing? **A**

Help Answers

- **Label printer for cassette tapes...** Try the June issue of Micro User, page 39. **A**

Monitor Information

- **Fujitsu-ME503**. After last month's comments about this monitor, we have had a lot of come-back. The trouble is that it is totally inconclusive! Some people say it is superb, clear as a bell with no problems, whereas others say it is very poor quality. In fact, I understand that Viglen are no longer supplying it for the Archimedes.

- **NEC Multi-sync II**. One dealer has apparently been telling prospective purchasers that "the scan rate output of the Archimedes can damage the Multi-sync II". I, personally, would doubt whether there was any truth in that claim but does anyone have the technical know-how to comment on it? **A**

Matters Arising...

• **24-pin Screen dump** – This is available to Archive subscribers for £5, as stated in the Software Available section last month, but NOT from Archive Magazine. Please send cheques to Abacus Computers.

• **Art-Worker**, the art package that was reviewed last month was from "McSoft", not "MacSoft". The latter is a company associated with the Apple Mac. Also, the proprietor, N. McNamara, asks that cheques should be made payable to him personally and not to McSoft.

• **Using the Debugger**. In the Memory01 program, line 60 should have an END on the end, i.e. 60 ON ERROR PROCerror:END

• **Improvements to Memory02**. Having had some feedback about this program, the two problems referred to in the last article, namely the failure of *MemoryI to display the inverted commas of an SWI instruction and the failure of the program to convert "ASC" to the token &97, seem to cause users more trouble than it takes to put right. Accordingly Gerald has written a new PROCassemble of about 40 lines which overcomes the problem. Readers can load their Memory02 and then type in the new lines from 25000 onwards. Those who buy the program disc will find Memory04 is the version with the new PROCassemble.

```
25000 DEF PROCassemble
25010 LOCAL character%, char%,
           operation$, operand$
25020 LOCAL ERROR
25030 ON ERROR LOCALPROCfloatingpoint
           :ENDPROC
25040 :
25050 REPEAT
25060 IF LEFT$(mnemonic$,1)=" "
           THEN mnemonic$=MID$(mnemonic$,2)
25070 UNTIL LEFT$(mnemonic$,1)<>" "
25080 FOR character%=1 TO INSTR
           (mnemonic$, " ")
25090 char%=ASC(MID$(mnemonic$,
           character%,1))
```

```
25100 IF char%>=60 AND char%<=7B THEN
25110 char%=char%&-&20
25120 MID$(mnemonic$,character%)
           =CHR$(char%)
25130 ENDIF
25140 NEXT character%
25150 operation$=LEFT$(mnemonic$
           ,INSTR(mnemonic$," "))
25160 operand$ = RIGHT$(mnemonic$,
           LEN(mnemonic$)-INSTR
           (mnemonic$," "))
25170 REPEAT
25180 IF LEFT$(operand$,1)=" " THEN
operand$=MID$(operand$,2)
25190 UNTIL LEFT$(operand$,1)<>" "
25200 IF INSTR(operand$,";") THEN
           operand$=LEFT$(operand$,
           INSTR(operand$,";")-1)
25210 operand$=operand$+" "
25220 operand$=LEFT$(operand$,INSTR
           (operand$," ") -1)
25230 IF LEFT$(operation$,3)="SWI"
           AND LEFT$(operand$,3)="OS_" OR
           LEFT$(operand$,4)="XOS_" THEN
25240 operand$="""+operand$+""""
25250 mnemonic$=operation$+" "+
           operand$+" ;"
25260 ENDIF
25270 :
25280 IF INSTR(mnemonic$,"ASC") THEN
25290 mnemonic$=LEFT$(mnemonic$,
           INSTR(mnemonic$,"ASC")-1)+CHR$(
           (&97)+MID$(mnemonic$,INSTR
           (mnemonic$,"ASC")+3)+" "
25300 ENDIF
25310 :
25320 $(TOP-&43)=LEFT$(mnemonic$,50)
25330 display%=TRUE
25340 :
25350 P%=start%+&20
25360 [OPT 0
25370 .store
25380 ;;;;;;;;;;;;;;;;;;;;;;;;;;
           ;;;;;;;;;;;;;;;;;;;;;;;;;;
25390 ]
25400 ENDPROC
```

• **What has happened to Memory03?** The "Memory03" version includes line 400 LIBRARY "&.BasicProgs.FpAss" and line 25030, is changed so that, instead of calling the rather limited floating point procedure PROCfloatingpoint, it calls the full BASIC floating point assembler contained in the LIBRARY program "FpAss" with `!(start%+&20)=FNassemble(mnemonic$)`. Version "Memory03" is a disassembler which can be used to enter not only ARM mnemonics but, through the routines contained in "FpAss", floating point mnemonics as well. Self-contained code generating LIBRARY functions which use lines 1350 to 1430 of the program "Memory03" can also contain floating point mnemonics. These floating point mnemonics are called, for example, with `FpAss("ATNE F0,#1.0 ; arctan(1) = pi/4")`, which places an extended precision value of $\pi/4$ in F0: the mnemonic will be assembled to the correct floating point code.

• **Disk Editor** (Archive Issue 6) Mr Terry Bromilow found a bug in the Disk Editor program. The program would sometimes save the changed information to the sector being moved to not the one being moved from! Oops! Anyway, here are the necessary changes: add:

```
2721 old_t=track
2722 old_h=head
2723 old_s=sector
2821 SWAP old_t,track
2822 SWAP old_h,head
2823 SWAP old_s,sector
2831 SWAP old_t,track
2832 SWAP old_h,head
2833 SWAP old_s,sector
2834 PROCnext
```

line 2830 should read:

```
2830 found=FALSE:PROCsave A
```

BASIC OVERLAY

Mention was made last month of (the as yet unreleased) version 1.03 of BASIC V and its improved TRACE performance. A further feature of the new interpreter is the provision of overlay facilities. The new keyword OVERLAY extends the function and procedure library provisions. In version 1.00 a library of functions and procedures may be INSTALLED in a program before it is run. Then if the original program calls a function or a procedure that it can not find, it searches the library. Using OVERLAY the search may be extended to disc. Libraries of functions and procedures may thus be loaded into a program as required while it is running. The reserved area of memory is overwritten each time a new library is loaded. Program size is then no longer limited by the machine RAM. In many respects, the large memory capacity of the Archimedes obviates the need for an OVERLAY facility, although its absence from version 1.00 was bemoaned by some reviewers. However, it does encourage well structured programming. In any case, it probably won't be too long before some users produce programs which exhaust the memory of even a 440! **A**

Small Ad's

DESKTOP GAMES for A310 upwards. Sliding blocks, mastermind, solitaire, *commands and volume control from desktop. With FREE hidden 100% m/c invaders game – can you find it? £5.95 inc. from GEM ELECTRONICS – see FactFile.

RADIOACTIVE ISOTOPES program showing decay chains, decay products, half-lives and abundances of 2000 odd nuclides. Tutorial Info / Edit mode. Requires A310 + multi-sync monitor. £5.95 from R.W. Darlington, 1 Fells Grove, Worsley, Manchester, M28 5JN. **A**

Sound Synthesis – Part 2

Ian Nicholls

Last month I began to explore the undocumented WaveSynth module and how it uses wavetables to create new voices. A listing of an Acorn program called "FMBrass14" was included to enable you to create a voice sounding something like a trumpet. I will return to FMBrass14 in next month's article because there are some extra bits in the original program, which I did not include, which suggest other useful features in WaveSynth, yet to be discovered. This month's article, however, concentrates on "FileGen" and "Arc Wave", two new programs which provide an easy means of generating new voices using the two main methods of 'additive harmonic synthesis' and 'frequency modulation synthesis'.

Fourier's Theorem

Before describing the programs themselves and how to use them, I need to spend a moment or two on a theorem developed by Fourier. What he was able to show was that any periodic (i.e. repetitive) waveform can be reproduced by adding together a series of sine waves. Each sine wave in the series has a frequency which is a whole number multiple of a basic frequency, called the fundamental, and any waveform can be synthesised by choosing appropriate multiples (harmonics) of the fundamental frequency and their relative amplitudes. Stated mathematically, any waveform, $w(t)$, can be expressed as:-

3D CAD/ANIMATION SYSTEM

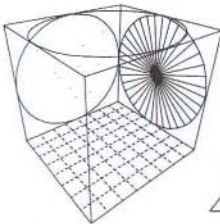
ARCHIMEDES SOFTWARE

Once again Silicon Vision steals the lead with this incredible offer. It's true we're offering our international, best-selling 3D Graphics Development System (3D GDS) at an incredibly low price. No we haven't gone mad, it's simply our way of introducing you to the world of 3D computer graphics.

3D GDS is a full blown 3D CAD & Animation system that can handle 3D models of any complexity. The package consists of a single disc providing the Design & Animation facilities and a library of predefined models to get you started. A comprehensive 95 page manual describes all the facilities including tutorials and example 3D animation programs on disc. The standalone 3D animation facilities take advantage of all the Archimedes screen, colour and plotting modes including dual screens for high speed flicker-free 3D animation - ideal for games & simulations.

Design facilities include 3D Editors to design objects with lines, create symmetrical objects by Sweeping simple sections, use objects as building blocks to create more complex objects which in turn may be used as building blocks, specify individual line colours for multi-coloured objects, dynamic 3D viewing from any angle and an applications facility to animate the 3D models from your own Basic or Machine code programs.

You'll find no better company to do business with than Silicon Vision. As Europe's fastest growing CAD software house, there's no-one more capable of satisfying your future needs. In the meantime we offer you 3D GDS with our compliments. Worth over £25 it can now be yours for only £19.95.



Compatible with Archimedes 305, 310, 410 and 440. Order one today and enter the dynamic world of high speed 3D computer graphics.

SPECIAL OFFER TO ARCHIMEDES MEMBERS ONLY (Quote Membership Number when Ordering).

3D CAD/ANIMATION SYSTEM: £17.95

SILICON VISION

All prices are fully inclusive.

Contact your local dealer or order directly by cheque/P.O./Access/Mastercard/Eurocard from:

SILICON VISION LTD, SIGNAL HOUSE, LYON RD, HARROW, MDDX, HA1 2AG.
Tel: 01-422 2274 or 01-861 2173. Fax: 01-427 5169. Telex: 918266 SIGNAL G.



$$w(t) = a_1 \sin(ft) + a_2 \sin(2ft) + a_3 \sin(3ft) + \dots + a_n \sin(nft) + \dots$$

where $f = 2 \times \pi$, t = time, a_n = the amplitude of the n th harmonic

The first method of creating new voices in our two programs uses the above concept: it allows you to build a sound waveform from its component sine waves (its harmonics) and is, therefore, called additive harmonic synthesis. In addition to specifying the harmonics, it also allows you to describe how the amplitude of each harmonic varies from the start of a note to the end. Much of the character of a particular instrument comes from the way in which the different harmonics of the fundamental note build up and decay.

The second method of creating new voices is frequency modulation synthesis. Again this relies on adding various harmonics to a fundamental note, but it does this in a different way. Basically what it does is to alter (or modulate) the frequency of the note whilst it is sounding. The waveform of a pure tone, such as that produced by a flute or recorder, is:-

$$w(t) = \sin(ft)$$

where $f = 2 \times \pi$ and t = time

In simple frequency modulation the waveform becomes:-

$$w(t) = a \sin(f_c t + i \sin(f_m t))$$

where a = the amplitude of the carrier wave
 f_c = the frequency of the carrier wave
 f_m = the frequency of the modulating wave
 i = the amplitude of the modulating wave
 t = time

In complex frequency modulation, two modulating waves are used instead of one:-

$$w(t) = a \sin(f_c t + i_0 \sin(f_{m0} t) + i_1 \sin(f_{m1} t))$$

The effect of frequency modulation is to add harmonics to the fundamental frequency; the larger the amplitude of the modulating wave, the

more harmonics there are created. The harmonics are both higher and lower in frequency than the carrier wave.

Many of you will not be too interested in the above, but I have included it because if you want to experiment with generating your own new voices it helps to understand what the different input parameters to the programs actually do!

Wavetable Data File Generator (FileGen)

This program creates data files for the second program, Arc_Wave, which then generates the wavetables for a particular voice. When FileGen is first run, it checks to see whether your disc has a directory called WTDDataFile on it, in which to store the wavetable data files. If not then it creates one. This task is accomplished using the OS File command with a parameter value of 5: this searches the disk for a defined object, returning 0 if it does not exist, 1 if it is a file and 2 if it is a directory.

You are then asked which of the three methods of creating a voice you want to use. I will describe the data that you need to input for each one in turn and list some examples as a basis for further experimentation.

Additive Harmonic Synthesis

When you choose this method, you will be asked to enter the following pieces of data:-

1. Name to be given to the new voice
– enter a name of up to 10 characters
2. Number of harmonics
– enter the number for your voice, between 1 and 20
3. Number of harmonic envelopes entered
– this keeps a running total of how many you have entered; it reads 0 to begin with
4. Amplitude envelope breakpoints for each harmonic
– enter the value of the harmonic itself
5. Time amplitude
– you now have to enter the shape of the amplitude envelope for the 3 chosen

harmonics by specifying pairs of values for time and amplitude: time is measured in centiseconds (and ranges from 0 to 500) and amplitude ranges from 0 to 1.0 – the end of the envelope is indicated by entering values of 0 for both parameters

– steps 4 and 5 are now repeated until the number of harmonics entered in step 1 is reached

6. Segment number for release phase

– each centisecond is called a segment and there is a separate wavetable for each segment. When a voice stops sounding, the sound does not cease immediately (unless a new note is played straight away), it decays over a short period of time. The number you enter here will be the segment from which WaveSynth starts this decay process, so it needs to be within, say, 50 segments (half a second) of the last segment.

I will take a voice called Clarinet to illustrate the process. The data entered (in bold) is:

Name:– **Clarinet**

No. of harmonics:– **3**

Amplitude breakpoints for each harmonic:–

Harmonic no.					
1		3		5	
<u>time</u>	<u>amp</u>	<u>time</u>	<u>amp</u>	<u>time</u>	<u>amp</u>
0	0	0	0	0	0
1	0.03	3	0.05	3	0
4	0.19	6	0.40	5	0.19
7	1.0	28	0.33	8	0.88
27	0.91	301	0.33	12	0.84
300	0.91	317	0.03	17	0.69
326	0	326	0	31	0.64
0	0	0	0	304	0.64
				311	0.33
				315	0.03
				326	0
				0	0

Segment number for start of release phase:– **300**

Simple FM Synthesis

The data to be entered is less complex than that for harmonic synthesis:–

1. Name – as for harmonic synthesis
2. Amplitude envelope breakpoints
– the same as for harmonic synthesis, but this time you do not have to enter the number of harmonics and envelope breakpoints for each one (since there is only one)
3. Modulation envelope breakpoints
– the format is the same as for the amplitude envelope breakpoints but, in this case, the modulation index (mod index) can have values between 0 and 25
4. Ratio of **carrier** frequency to the fundamental
– the fundamental has a frequency of 1, and usually the carrier will be the same, so the value entered will be 1
5. Ratio of the **modulator** frequency to the fundamental
– varying this value (it must be positive) will alter the harmonics generated and, hence, the nature of the resulting sound (it takes values between 0 and 25)
6. Segment number for release phase
– as for harmonic synthesis.

We can illustrate Simple FM Synthesis with the data from last month's article for the "FMBrass14" voice:

Name:– **Brass14**

Amplitude envelope breakpoints:–

<u>time</u>	<u>amplitude</u>
0	1.0
1	1.0
30	0.8
300	0.8
330	0
0	0

Modulation envelope breakpoints:–

<u>time</u>	<u>mod index</u>
0	3.0
5	7.0

<u>time</u>	<u>mod index</u>
30	4.5
300	6.5
330	0
0	0

Carrier frequency ratio:— 1

Modulator frequency ratio:— 1.001

Segment number for start of release phase:— 300

Complex FM Synthesis

The format for the data for this form of synthesis is similar to that for Simple FM, except that you will have to enter two sets of modulation envelope breakpoints, one for each modulator and you will have to specify the “modulator to carrier frequency ratio” for two modulators and not one. I will illustrate this method with data to create a piano-like voice.

Name:— **Piano**

amplitude envelope breakpoints:—

<u>time</u>	<u>amplitude</u>
0	0
2	0.95
3	1.0
20	0.60
40	0.30
90	0.15
170	0.07
285	0.02
300	0
0	0

modulator 1 envelope breakpoints:—

<u>time</u>	<u>mod index</u>
0	1
300	1
0	0

modulator 2 envelope breakpoints:—

<u>time</u>	<u>mod index</u>
0	0.2
300	0.2
0	0

Carrier frequency ratio:— 1

Modulator 1 frequency ratio:— 1

Modulator 2 frequency ratio:— 4

Segment number for release phase:— 285

When you have entered all of the data for a new voice, FileGen will list all of the wavetable data files that have now been stored in directory WTDataFile. It will then asks you whether you want to create another data file, build a wavetable from an existing data file or exit. If you choose the second option, the Arc_Wave program is CHAINED.

Arc_Wave – Wavetable Voice Builder

This is the program that does the real work: it generates the set of wavetables for a particular voice from a data file created with FileGen. It begins by listing the data files (held in directory WTDataFile) and asks you which one to use. It then tells you how many segments there will be in your wavetable and counts, on screen, as each segment is calculated. This process takes quite a few minutes, particularly for additive harmonic synthesis, where two passes through the whole set of data are necessary in order to scale the maximum amplitude of the waveform to 1.0.

When the calculations are complete, the program tells you at which segment the release phase starts and then ensures that a WaveTables directory exists on your disc. You are then asked for a name for the voice and the set of wavetables is stored in WaveTables with this name.

Lastly, you are given the option of building another voice using a further data file, returning to FileGen to create more data files, or exiting.

Here is the data for two more voices, both use additive harmonic synthesis:—

Name:— **Organ**

No. of harmonics:— 5

Amplitude breakpoints for each harmonic:—

Harmonic number					
1		2		4	
<u>time</u>	<u>amp</u>	<u>time</u>	<u>amp</u>	<u>time</u>	<u>amp</u>
0	0	0	0	0	0
3	1.0	3	0.7	3	0.4
6	0.8	6	0.55	6	0.3
250	0.8	250	0.55	250	0.3
300	0	300	0	300	0
0	0	0	0	0	0

Harmonic number			
6		8	
time	amp	time	amp
0	0	0	0
3	0.2	3	0.1
6	0.15	6	0.08
250	0.15	250	0.08
300	0	300	0
0	0	0	0

Segment number for start of release phase:—250

Name:— **Trumpet**

No. of harmonics:— 8

Amplitude breakpoints for each harmonic:—

Harmonic number					
1		2		3	
time	amp	time	amp	time	amp
0	0	0	0	0	0
4	0.8	4	0.82	6	0.2
5	0.9	5	0.96	7	1.0
250	0.9	250	0.96	250	1.0
260	0.75	258	0.77	254	.95
270	0.25	265	0.1	265	0.1
280	0	275	0	270	0
0	0	0	0	0	0

Harmonic number					
4		5		6	
time	amp	time	amp	time	amp
0	0	0	0	0	0
6	0.65	4	0.15	7	0.2
250	0.65	5	0.33	9	0.25
258	0.58	250	0.33	254	0.25
264	0.1	257	0.3	268	0.05
270	0	268	0.05	270	0
0	0	270	0	0	0
		0	0		

Harmonic number			
7		8	
time	amp	time	amp
0	0	0	0
7	0.1	6	0.06
9	0.18	8	0.16
250	0.18	250	0.16

time	amp	time	amp
255	0.20	255	0.15
263	0.05	263	0.04
268	0	268	0
0	0	0	0

Segment number for start of release phase:—250

Space has run out this month, so I will have to leave to the next article some further comments on the above programs and the wavetables they create. I will also give you the missing parts of Acorn's program from the first article and leave you with a series of questions about WaveSynth which those missing parts suggest! In the meanwhile, if you produce any interesting sounds of your own with FileGen and Arc_Wave, why not share them with the rest of us! (*We could even do a "Sound Shareware Disc". Ed.*)

```

10 REM > FileGen
20 REM Program Wavetable Data File
   Generator
30 REM Author Ian Nicholls
40 REM Version 1.00
50 REM ARCHIVE June 1988
60 REM Program Subject to Copyright
70 :
80 ON ERROR PROCerror:END
90 DIM time(20,29),amp(20,29),
   harm(20),bp%(20),buffer% 12
100 MODE 0:OFF
110 COLOUR129:COLOUR0:VDU28,18,
   4,59,0:CLS:VDU26
120 PRINTTAB(20,2)"ARCHIVE
   WAVETABLE DATA FILE GENERATOR"
130 COLOUR128:COLOUR1
140 $buffer%="WTDDataFile"
150 SYS"OS_File",5,buffer% TO
   file_type%
160 IF file_type%=0 THEN
170 *CDIR WTDDataFile
180 PRINTTAB(2,7)"Directory
   ""WTDDataFile"" has been created to
   hold your wavetable data files"
190 ENDIF
200 *DIR WTDDataFile
210 PRINTTAB(25,11)"TYPE OF
   SYNTHESIS REQUIRED:—"
220 PRINTTAB(25,16)"1 - ADDITIVE
   HARMONIC"

```

Sound Synthesis

```

230 PRINTTAB(25,18)"2 - SIMPLE FM"
240 PRINTTAB(25,20)"3 - COMPLEX FM"
250 REPEAT:input$=GET$:UNTIL
      INSTR("123",input$)<>0
260 type%=VAL(input$):CLS
270 ON:INPUTTAB(18,15)"Name to be
      given to the new voice ....
      "name$:OFF
280 channel%=OPENOUT(name$)
290 PRINT#channel%,name$,type%
300 CLS
310 CASE type% OF
320   WHEN 1:PROCharmonic
330   WHEN 2:PROCsimple_FM
340   WHEN 3:PROCcomplex_FM
350 ENDCASE
360 CLS:ON:INPUTTAB(18,14)"Enter
      segment number for start of
      RELEASE"TAB(18,17)"phase of
      the amplitude envelope ....
      "endseg%:OFF
370 PRINT#channel%,endseg%
380 CLOSE#channel%
390 OSCLI("SETTYPE "+name$+" FFD")
400 OSCLI("STAMP "+name$)
410 CLS:PRINTTAB(0,2)"The wavetable
      data files currently held in
      directory WTDataFile are
420 *UP          :-''':*CAT
430 PRINT''':TAB(19)"Please enter
      your choice:--"
440 PRINT''':TAB(19)"1 - create
      another wavetable data file"
450 PRINTTAB(19)"2 - build a
      wavetable from a data file"
460 PRINTTAB(19)"3 - exit"
470 REPEAT:input$=GET$:UNTIL
      INSTR("123",input$)<>0
480 CASE input$ OF
490   WHEN "1":RUN
500   WHEN "2":CHAIN"Arc_Wave"
510   WHEN "3":END
520 ENDCASE
530 :
540 DEFPROCCharmonic
550 CLS:PRINTTAB(0,0)"ADDITIVE
      HARMONIC SYNTHESIS"TAB(0,1)
      STRING$(27,"-")
560 PROCinput_value(0,3,"Number of
      harmonics - ",1,20,temp)
      :n_har%=INT(temp+0.01)
570 PRINTTAB(30,3)"number of
      harmonic envelopes entered - 0"
580 title$="amplitude envelope
      breakpoints for harmonic -
      ":bp%()=0
590 FOR I%=0 TO n_har%-1
600   PROCinput_value(0,5,title$
      ,0,20,harm(I%))
610   PRINTTAB(0,6);STRING$(LEN
      (title$)+2,"-")
620   PROCuniversal_input
      ("amplitude",1)
630   PRINTTAB(69,3);I%+1
640   VDU28,0,31,79,10:CLS:VDU26
650 NEXT
660 PRINT#channel%,n_har%
670 FOR I%=0 TO n_har%-1
680   PRINT#channel%,harm(I%)
690   FOR J%=0 TO bp%(I%)-1
700     PRINT#channel%,time(I%,J%)
      ,amp(I%,J%)
710 NEXT:NEXT
720 ENDPROC
730 :
740 DEFPROCinput_value(x%,y%,text$,
      low_lim,up_lim,RETURN value)
750 length%=LEN(text$)
760 REPEAT
770   PRINTTAB(x%+length%,y%)" "
780   PRINTTAB(x%,y%)text$+"?"
790   INPUTTAB(x%+length%,y%)value
800 UNTIL (value>=low_lim AND
      value<=up_lim)
810 ENDPROC
820 :
830 DEFPROCuniversal_input(string$
      ,upper)
840 FOR J%=0 TO 2
850   PRINTTAB(18*J%,8)"time "+
      string$TAB(18*J%,9)"---- ";
      STRING$(LEN(string$),"-")
860 NEXT
870 K%=-1:REPEAT:K%=K%+1
880   PROCinput_value(2-18*(K%>9)-
      18*(K%>19),10+K%*2+20*(K%>9)+20
      *(K%>19),"",0,500,time(I%,K%))
890   PROCinput_value(8-18*(K%>9)-
      18*(K%>19),10+K%*2+20*(K%>9)+20*
      (K%>19),"",0,upper,amp(I%,K%))
900   bp%(I%)+=1
910 UNTIL (K%>0 AND time(I%,K%)=0
      AND amp(I%,K%)=0)

```

```

920 ENDPROC
930 :
940 DEFPROCsimple_FM
950 CLS
960 PRINT TAB(0,0)"SIMPLE FM
      SYNTHESIS" TAB(0,1) STRING$
      (19,"-")
970 PRINTTAB(0,5)"amplitude
      envelope breakpoints"
980 PRINTTAB(0,6);STRING$(30,"-")
990 I%=0:bp%=0:PROCuniversal_
      input("amplitude",1)
1000 FOR J%=0 TO bp%(0)-1
1010   PRINT#channel%,time(0,J%),
      amp(0,J%)
1020 NEXT
1030 VDU28,0,31,79,4:CLS:VDU26
1040 PRINTTAB(0,5)"modulation
      envelope breakpoints"
1050 PRINTTAB(0,6);STRING$(31,"-")
1060 I%=0:bp%=0:PROCuniversal_
      input("mod index",25)
1070 FOR J%=0 TO bp%(0)-1
1080   PRINT#channel%,time(0,J%),
      amp(0,J%)
1090 NEXT:CLS
1100 INPUTTAB(10,14)"Ratio of
      CARRIER frequency to that of the
      fundamental - "FC
1110 INPUTTAB(10,16)"Ratio of
      MODULATOR frequency to that of the
      fundamental - "FM1
1120 PRINT#channel%,FC,FM1
1130 ENDPROC
1140 :
1150 DEFPROCcomplex_FM
1160 CLS
1170 PRINT TAB(0,0)"COMPLEX FM
      SYNTHESIS" TAB(0,1) STRING$
      (20,"-")
1180 PRINTTAB(0,5)"amplitude
      envelope breakpoints"
1190 PRINTTAB(0,6);STRING$(30,"-")
1200 I%=0:bp%=0:PROCuniversal_
      input("amplitude",1)
1210 FOR J%=0 TO bp%(0)-1
1220   PRINT#channel%,time(0,J%),
      amp(0,J%)
1230 NEXT
1240 VDU28,0,31,79,4:CLS:VDU26
1250 PRINTTAB(0,5)"modulation
      envelope breakpoints for
      modulator - "

1260 PRINTTAB(0,6);STRING$(49,"-"):
      bp%=0
1270 FOR I%=0 TO 1
1280   PRINTTAB(49,5);I%+1
1290   PROCuniversal_input("mod
      index",25)
1300   FOR J%=0 TO bp%(I%)-1
1310     PRINT#channel%,time(I%,J%),
      amp(I%,J%)
1320   NEXT
1330   VDU28,0,31,79,10:CLS:VDU26
1340 NEXT
1350 CLS
1360 INPUTTAB(10,13)"Ratio of
      CARRIER frequency to that of
      the fundamental - "FC
1370 INPUTTAB(10,15)"Ratio of
      MODULATOR 1 frequency to that
      of the fundamental - "FM1
1380 INPUTTAB(10,17)"Ratio of
      MODULATOR 2 frequency to that
      of the fundamental - "FM2
1390 PRINT#channel%,FC,FM1,FM2
1400 ENDPROC
1410 :
1420 DEFPROCerror
1430 CLS:REPORT:PRINT" at line ";ERL
1440 CLOSE#0:*UP
1450 ENDPROC

10 REM > Arc_Wave
20 REM Program Wavetable Voice
      Builder
30 REM Author Ian Nicholls
40 REM Version 1.00
50 REM ARCHIVE June 1988
60 REM Program Subject to Copyright
70 :
80 ON ERROR PROCerror:END
90 MODE0:OFF
100 VDU28,21,4,57,0
110 COLOUR129:CLS:VDU26
120 COLOUR0:PRINTTAB(24,2)"ARCHIVE
      WAVETABLE VOICE BUILDER"
130 COLOUR1:COLOUR128:PROCdelay(200)
140 PRINTTAB(4,8)"The wavetable
      data files currently held in
      directory ""WTDataFile"" are:-
      ""':*DIR WTDataFile
150 PROCdelay(200):*CAT
160 vert%=VPOS+3:PROCdelay(200)

```

Sound Synthesis

```

170 ON:INPUTTAB(20,vert%) "Please
    type in the name of the wavetable"
    TAB(20,vert%+1) "data file
        to be used .... "name$
180 OFF:channel%=OPENIN(name$)
    :PROCdelay(150)
190 :
200 PROCinitialise
210 CASE type% OF
220   WHEN 1:
230     INPUT#channel%,n_har%
240     PROCread_amp_brkpts
250     PROCbuffer_setup
260     PRINTTAB(0,3) "FIRST PASS"
270     t%=0
280     WHILE t% < MaxEndTime
290       PROCamp_segment1
300       PROCamp_segment2
310       t%+=1
320     ENDWHILE
330     t%=0
340     PRINTTAB(0,7) "SECOND PASS"
350     WHILE t% < MaxEndTime
360       PROCend_of_segment
370       WaveTable%?t%=FNDAClog
         (TableEntry(t%)/MaxTableEntry)
380       t%+=1
390     ENDWHILE
400     WHEN 2,3:
410       n_har%=1
420       PROCread_amp_brkpts
430       PROCread_mod_brkpts
440       PROCbuffer_setup
450       t%=0
460       WHILE t% < MaxEndTime
470         PROCamp_segment1
480         PROCmod_segment
490         PROCend_of_segment
500         p=((t%MOD256)+0.5)*z
510         IF type%=2 THEN
520           x=a(0)*SIN(FC*p+i(0)
             *SIN(FM1*p))
530         ELSE
540           x=a(0)*SIN(FC*p+i(0)*SIN
             (FM1*p)+i(1)*SIN(FM2*p))
550         ENDIF
560         l%=FNDAClog(x)
570         WaveTable%?t%=l%
580         t%+=1
590       ENDWHILE
600     ENDCASE
610     INPUT#channel%,endseg%

620 PRINT"" "The RELEASE phase of the
    amplitude envelope starts at
        segment ";endseg%
630 CLOSE#channel%:PROCdelay(200):*UP
640 $buffer%="WaveTables"
650 SYS"OS_File",5,buffer% TO
        file_type%
660 IF file_type%=0 THEN
670   *CDIR WaveTables
680   PRINT"Directory ""Wave
        Tables"" has been created to
        hold your wavetable files"
690 ENDIF
700 WaveDescriptor%!0=0 :REM end
        waveform
710 WaveDescriptor%!4=0
720 PROCheader
730 PROCsave
740 CLS:PRINT"" "TAB(17) "Please
        enter your choice:-"
750 PRINT"" "TAB(17) "1 - build a
        wavetable from another data file"
760 PRINT"TAB(17) "2 - create
        further wavetable data files"
770 PRINT"TAB(17) "3 - exit"
780 REPEAT:input$=GET$:UNTIL INSTR
        ("123",input$)<>0
790 CASE input$ OF
800   WHEN "1":RUN
810   WHEN "2":CHAIN"FileGen"
820   WHEN "3":VDU26:END
830 ENDCASE
840 :
850 DEFPROCinitialise
860   NBkPts% =30
870   NHarmonics%=20
880   MaxTableEntry=0
890 :
900   DIM AmpEnd(NBkPts%-
        1,NHarmonics%-1),AmpTime
        (NBkPts%-1,NHarmonics%-1)
910   DIM ampBP%(NHarmonics%-1),
        EndTime(NHarmonics%-1)
920   DIM ModEnd(NBkPts%-1,1),ModTime
        (NBkPts%-1,1),modBP%(1),
        ModEndTime(1)
930   DIM a(NHarmonics%-1),
        A(NHarmonics%-1),
        AT(NHarmonics%-1),dA(NHarmonics%-
        1),harm(NHarmonics%-1)
940   DIM i(1),I(1),IT(1),dI(1),
        buffer% 12

```

```

950 z=PI/128
960 INPUT#channel%,name$,type%
970 VDU28,0,31,79,8:CLS
980 ENDPROC
990 :
1000 DEFPROCread_amp_brkpts
1010 FOR I%=0 TO n_har%-1
1020   bp%=0
1030   IF type%=1 INPUT#channel%,
                        harm(I%)
1040   INPUT#channel%,t,v
1050   AmpTime(bp%,I%)=FNTime(t)
1060   AmpEnd(bp%,I%)=v
1070   REPEAT
1080     INPUT#channel%,t,v
1090     t = FNTime(t)
1100     IF t>AmpTime(bp%,I%) THEN
1110       bp%+=1
1120       AmpTime(bp%,I%)=t
1130       AmpEnd(bp%,I%)=v
1140     ENDIF
1150   UNTIL t < AmpTime(bp%,I%)
1160   EndTime(I%) = AmpTime(bp%,I%)
1170 NEXT
1180 MaxEndTime=0
1190 FOR I%=0 TO n_har%-1
1200   IF EndTime(I%) > MaxEndTime
      MaxEndTime=EndTime(I%)
1210 NEXT
1220 DIM TableEntry(MaxEndTime)
1230 ENDPROC
1240 :
1250 DEFPROCread_mod_brkpts
1260 IF type%=2 THEN n_mod%=0 ELSE
      n_mod%=1
1270 FOR I%=0 TO n_mod%
1280   bp%=0
1290   INPUT#channel%,t,v
1300   ModTime(bp%,I%)=FNTime(t)
1310   ModEnd(bp%,I%)=v
1320   REPEAT
1330     INPUT#channel%,t,v
1340     t = FNTime(t)
1350     IF t>ModTime(bp%,I%) THEN
1360       bp%+=1
1370       ModTime(bp%,I%)=t
1380       ModEnd(bp%,I%)=v
1390     ENDIF
1400   UNTIL t < ModTime(bp%,I%)
1410   ModEndTime(I%)=ModTime(bp%,I%)
1420 NEXT
1430 INPUT#channel%,FC,FM1

1440 IF type%=3 THEN INPUT#channel%
                        ,FM2
1450 ENDPROC
1460 :
1470 DEFPROCbuffer_setup
1480 nWaves%=MaxEndTime DIV 256
1490 PRINT"Your wavetable data file
      will create a wavetable"
      "consisting of ";nWaves%;" segments"
1500 WaveSize%=nWaves%
1510 WaveHeader%=(64 + (nWaves%+1)
      *(8)+255)DIV256
1520 BuffSize%=(WaveHeader%
      +WaveSize%)*256
1530 DIM Buff% BuffSize%-1
1540 WaveTable% = Buff%+WaveHeader%
      *256
1550 WaveDescriptor% = Buff%+64
1560 descriptor%=8
1570 ENDPROC
1580 :
1590 DEFPROCamp_segment1
1600 FOR I%=0 TO n_har%-1
1610   IF t% > EndTime(I%) THEN
1620     a(I%)=0
1630   ELSE
1640     IF t%>=AmpTime(ampBP%(I%)
      ,I%) THEN
1650       A(I%)=AmpEnd(ampBP%(I%)
      ,I%)
1660       AT(I%)=AmpTime(ampBP%
      (I%),I%)
1670       ampBP%(I%)+=1
1680       dA(I%)=(AmpEnd(ampBP%(I%)
      ,I%)-A(I%))/(AmpTime
      (ampBP%(I%),I%)-AT(I%))
1690     ENDIF
1700     a(I%)=A(I%)+dA(I%)*(t%-
      AT(I%))
1710   ENDIF
1720 NEXT
1730 ENDPROC
1740 :
1750 DEFPROCamp_segment2
1760 p=((t%MOD256)+0.5)*z
1770 x=0
1780 FOR I%=0 TO n_har%-1
1790   x+=a(I%)*SIN(p*harm(I%))
1800 NEXT
1810 TableEntry(t%)=x
1820 IF x>MaxTableEntry THEN
      MaxTableEntry=x

```

Sound Synthesis

```

1830 IF (t%MOD256)=0 PRINTTAB(0,4)      2200 $Buff%=title$
      ;"Segment: ";t%DIV256            2210 Buff%?(LENTitle$)=0
1840 ENDPROC                             2220 Buff%!16=BuffSize%
1850 :                                  2230 Buff%!20=8
1860 DEFPROCmod_segment                 2240 Buff%!24=8
1870 FOR I%=0 TO n_mod%                 2250 Buff%!28=8
1880   IF t% > ModEndTime(I%) THEN      2260 Buff%!32=8
1890     i(I%)=0                        2270 Buff%!36=8
1900   ELSE                             2280 Buff%!40=8
1910     IF t%>=ModTime(modBP%(I%)      2290 Buff%!44=8
      ,I%) THEN                        2300 Buff%!48=8
1920       I(I%)=ModEnd(modBP%(I%)      2310 Buff%!52=8+endseg%
      ,I%)                             2320 ENDPROC
1930       IT(I%)=ModTime(modBP%(I%)    2330 :
      ,I%)                             2340 DEFPROCsave
1940       modBP%(I%)+=1                2350 OSCLI("SAVE WaveTables."
1950       dI(I%)=(ModEnd(modBP%(I%)    +name$+" "+STR$~Buff%+" "+STR$~
      ,I%)-I(I%))/(ModTime(modBP%      BuffSize%+" 0 FFFFD00")
      (I%),I%)-IT(I%))                 2360 OSCLI("STAMP WaveTables."+name$)
1960   ENDIF                             2370 ENDPROC
1970   i(I%)=I(I%)+dI(I%)*(t%-         2380 :
      IT(I%))                          2390 DEF FNTIME(t) = t*256:REM
1980   ENDIF                             convert c-secs to sample periods
1990 NEXT                                2400 :
2000 ENDPROC                             2410 DEFFNDAClog(N)
2010 :                                  2420 REM log of N in the range +/- 1
2020 DEFPROCend_of_segment              2430 LOCAL S%,N% : IF N < 0 THEN
2030 IF (t%MOD256)=0 THEN                S%=&01: N=ABS(N) ELSE S%=&00
2040   PRINTTAB(0,8);"Segment:         2440 IF N=0 THEN = S%
      ";t%DIV256;"                      2450 N%=0.5+INT(N*4096+16)
2050   WaveDescriptor%!0=&FF +          2460 IF N% > &FFF THEN = S% OR &FE
      (51<<9) :REM one waveform        2470 WHILE N%>=32
2060   WaveDescriptor%!4=                2480   N%=N%>>1 : S%=S%+&20
      ((descriptor%+1)<<16)+WaveHeader% 2490 ENDWHILE
      +(t%DIV256)                       2500 =S% OR ((N%AND&F)*2)
2070   WaveDescriptor%+=8               2510 :
2080   descriptor%+=1                   2520 DEFPROCdelay(time%)
2090 ENDIF                             2530 TIME=0:REPEAT UNTIL TIME>time%
2100 ENDPROC                             2540 ENDPROC
2110 :                                  2550 :
2120 DEFPROCheader                       2560 DEFPROCerror
2130 REPEAT                             2570 CLS:REPORT:PRINT;" at line ";ERL
2140   CLS:ON:INPUT" What name do      2580 CLOSE#0:*UP
      you wish to give to this         2590 ENDPROC A
      wavetable ? "name$
2150 UNTIL LENname$>0 AND LENname$
      <12:OFF
2160 FOR A%=0 TO 63 STEP 4
2170   Buff%!A%=0
2180 NEXT
2190 title$="!WT:"+name$

```

PAINT YOURSELF A PROGRAM

If you're a programmer looking for an easy way to incorporate sophisticated graphics into your programs, ArcIEd is the perfect answer.

ArcIEd is a comprehensive graphics editor, written specially for the Archimedes, which allows you to draw pictures and store them. Not as pictures – but as programs!

Using the mouse, pop-up menus and the function keys, you can build up your picture from solid lines, dotted lines, circles, ellipses, triangles, rectangles, parallelograms...

A user-definable grid (a sort of on-screen graph paper) helps you square up your drawing and add text.

A 'snap' (join to the closest existing point) facility makes sure that your figures are closed and your fills don't flood out.

Colours are selected from and mixed on a colour menu.

And there are built-in transformations – reflection, rotation, translation and scaling. All of them can be applied to the full screen, or to a user-defined window.

And because while you're drawing you're also writing a program, you can edit your picture, stepping backwards and forwards through the drawing, modifying whatever you want to change. Both versions remain onscreen until you ask the computer to redraw the picture, so you can see exactly what you're doing.

Once your picture's complete, you can either save it to disk or to memory as a BASIC program. And to ensure that the routine will fit straight into an existing program, you can specify – as you save – whether you want to retain existing colour statements, whether you want to save co-ordinates as absolute or relative, and whether you want the picture saved as a procedure with built-in parameters.

The whole package is fully documented in a unique online manual that you can call up onscreen at any time. You can either step through it, page by page, or call up the index and use the mouse to select the topic you want: the correct page will appear onscreen.

ArcIEd costs £24.95, including VAT, postage and packing.

GIVE YOUR ARC A HAND

Servo, specially designed for the Archimedes, allows you to control up to four servos (the type used in radio control cars and aircraft) from your computer.

In effect, it gives your Archimedes up to four 'hands' that it can use – with a little mechanical assistance – to control outside objects like switches, locks, doors, windows...

Servo consists of a small hardware unit which fits on the Archimedes' printer port, a software module and an 'online manual'.

You supply the servos (and the batteries which your servo needs).

The online manual explains how servos work, shows you how to wire them up, and provides an interactive test environment – so you can check that everything's working properly and you can experiment with test values to see what suits your particular servos.

Servo costs £29.95 including VAT, postage and packing.

Please send me:

☐ copy/copies of
ArcIEd at £24.95
(ArcIEd will be
despatched by return
post).

☐ copy/copies of
Servo at £29.95 (Please
allow 28 days for
delivery).

I enclose a cheque for £
made payable to Jansons.

Name

Address

Postcode Tel

Jansons

93 North Street,
Stilton,
PETERBOROUGH
PE7 3RR

FORTRAN 77 Graphics & Op Sys Library

Gwyneth Pettit (Academic Software)

As reported in the P.S. to last month's review of CCD's FORTRAN library, the system has been extended. The FORTRAN library still contains the CCD Graphics library file GxLib, but the VDU calls are now in a separate library as GvLib. You also get, in addition, the FORTRAN library we have all been waiting for, OsLib. The icing on the cake is that it costs no more than before and this system (for which the manual has been fully rewritten) would now be hard to fault – there is not much it doesn't do.

There are some more hidden extras, as well, such as the Acorn RS423 patch module (by permission of Acorn Computers Ltd.) in case you are plotting via a serial connection. You also have an explicit option for the Plotmate A3M plotter (from Linear Graphics Ltd.) as well as the original Hewlett-Packard plotter option, and the manual has been extensively rewritten to describe the wiring and configuration details of the interfacing of these devices.

OSLIB at last

If you are not an old hand at FORTRAN, you may well be asking "why all the fuss about OsLib?" "but, as I hinted in last month's review, this really is the sliced bread of the FORTRAN world, giving the sort of access to, and control over, the Arthur operating system which C programmers are already enjoying. Once again, users who are not previously familiar with the BBC Model B may feel out of the secret, but a review is not the place to learn about OSWORD and INKEY and I hope they will bear with the strange terms.

First, there are some really constructive low-level functions giving access to memory addresses of integer and character variables. They were necessary for the more obviously exciting subroutines and functions in OsLib, and are made available to the programmer so that,

for instance, memory areas can be set up for relocatable machine code subroutines – and there is a function IUSR provided to call them.

Just like BASIC !

The interface to the operating system is achieved by such functions as ISWI (for calling SWI routines), IOSCLI (for * commands), IOSBYTE, IOSWORD, IOSFILE, IOSARGS, IOSGBPB and IOSFIND. These give you access to a comprehensive range of effects quite unusual in a FORTRAN environment. There are routines to read and set the clock; there is IADVAL (not for Analogue/Digital conversion, unless the converter module is fitted, but for interrogating various buffers); there are the functions ICGET (like GET) and INKEY for single-character input from the keyboard; there is a subroutine MOUSE, to return mouse coordinates and button status; and there is a subroutine CPOS (like POS and VPOS in BASIC). What more could any graphics programmer want? (Don't tell me – tell CCD!).

Sound and WIMP

The library manual points out that by using IOSCLI and ISWI, the Archimedes sound facilities and the WIMP interface can be accessed, but cautions you that some of the WIMP SWI's have large parameter blocks, for which FORTRAN is not ideal, although the low-level routines provided here do make it feasible.

Programming Example

A sample of the sort of interactive program made possible by OsLib, this short program was written using XED (the text editor published by CCD Computer Services – see below.)

```
PROGRAM FORST
IPT=IOSCLI('MOUNT 0')
IPT=IOSCLI('CAT')
```

```
C
C This catalogues the floppy disk
C
```

```
CALL GVMODE(15)
DO 10 ICOUNT=1,100
  IPT=IOSCLI('SOUND 1 &FFFA
              150 25')
```

C
C Not quite the same as the old
SOUND command!

```
C
5  ICHAR=INKEY(100)
   IF (ICHAR.EQ.-1) GOTO 5
   IPT=IOSCLI('FX 15')
   ICHAR1=ICGET()
   CALL GVGCOL(3,ICOUNT)
   CALL GVMOVE(ICOUNT*10,ICHAR)
   CALL GVPLT(153,ICHAR1*5,0)

10  CONTINUE

C
C Plots coloured circles when
   two keys are pressed
```

C
END

(CCD Computer Services, price £49.50.) **A**

Micro-Aid

25 Fore Street, Praze, Camborne,
Cornwall, TR14 0JX.

Tel: 0209-831274

Extended Payroll – £57.44

ACCOUNT-PLUS – £57.44

Mailing – £34.44

Family History System – £24.95

135 tpi double sided 3.5" discs – £14.95 for 10

2000 sheets perf/tractor listing paper – £13.45

Dyneer DW20 14" DaisyWheel Printer – £225.00

2 x 2000 Mailing Labels – £9.95

Please add 57p Postage

The XED Text Editor

Gwyneth Pettit

If you are intending to write FORTRAN source code, you should certainly buy a copy of XED at £19.50 – a word-processor is not ideal for text editing (or vice-versa) so you probably need both. XED is workman-like and user-friendly, with most of the facilities of established mainframe text-editors (the author acknowledges its similarity to ZED). It provides both line-edit and full-screen modes and is entered via the command screen, where you can obtain a list of command words by pressing <print>. You call full-screen mode by typing the command 'fs' and you return to the command screen by pressing <escape> (like View or WordWise). This is more convenient than the TURBO editor of Turbo-Pascal which has no on-screen help and for which you need to have the manual by your side.

If you use XED frequently, you can set up command macros; you have full access to the operating system * commands; you can control it with the mouse and function keys to minimise typing; the manual is clear and well written and we could only find one omission on our short 'test run' – directing a file (or part of it) to a printer, we found required us to type, from within the command mode :-

OUTPUT [(start line) (end line)] PRINTER:

Conclusion

If you have bought the FORTRAN 77 compiler and expect to use it to write graphics programs, then you can confidently buy this Operating System and Graphics Library, because it will certainly save you much time and effort and your programs can exploit the Archimedes to the full. If you have already bought the original Graphics Library, CCD say they will upgrade your old disk free of charge. This Library, with its highly commendable manual, has to be one of this month's Best Buys.

(CCD Computer Services, price £19.50.) **A**

ISO Pascal on the Archimedes

Ian Smith & Leslie Wiggins

(Leicester Polytechnic)

Ian and Leslie have written some library routines, which they are selling, for ISO Pascal, so although this is an article kindly donated by them, please note that it is not intended to be a review of the products they are offering.

Pascal is a language commonly used to teach structured programming at Advanced and Undergraduate level. ISO Pascal for the Archimedes is a full and extended Level 1 implementation of the language. It comes on just one disk with a reasonable manual comprising implementation and configuration notes. It does not aim to provide material to enable a user to teach him/herself Pascal; numerous texts are available to do this.

The suggested configuration given in the manual is to copy the Compiler and Linker on to 2 separate disks. One disk would contain the compiler, editor (Twin is suggested but not part of the package) and also the programmer's source code. The linker, Pascal library and executable code would be on the other. Both will need to contain the partly compiled code produced by the compiler in aof (Acorn Object Format) for use by the linker. These will need to be copied from compiler disk to linker disk unless two drives are available. It is perfectly possible to manage with one disk as supplied but larger programs will require the use of 2 disks. Disk management could become a problem but the suggestion of keeping directories such as "pas" for source code and "lib" for library is clearly the right one.

It is not our aim in this article to evaluate the compiler and compare it with others available under different operating systems, suffice it to say that the compiler is fast and the linker reasonable, although the executable files it produces are rather large. It should, however, be

possible to reduce this by judicious use of the available compiler options. Aliases can be created to shorten the commands needed to compile and link files; some useful examples are given in the manual.

Acorn's implementation provides some useful extensions to the Pascal standard. One is the ability to include in-line assembler code within Pascal; this makes calls to the operating system possible and straightforward giving access to ALL Arthur code. A further extension which is equally, if not more, useful is the provision for "modules". This allows for the construction and compilation of separate blocks of code which can then be linked with any 'client' program. This is a powerful tool in program design and development. The Procedures and Functions in a Module together with any variables may be EXPORTED from the Module and IMPORTED by the 'client' program.

We wanted to produce 'useful' extensions to Acorn's ISO Pascal, via Modules suitable for General use, Graphics Operating System calls and WIMPS. After being delayed by the inadequacy of the pre-release version of Pascal and the limitations of OS 0.2, these Modules have now written and tested.

There are 5 modules available at present:

GENERAL – 21 routines such as CLS, VDU, MODE etc. (See below for details)

GRAPHICS – 36 routines such as PLOT, GCOL, RECTANGLE, MOUSE etc.

SPRITES – 15 routines such as SLOAD, SFLIP, SGET etc.

STRINGS – 12 routines such as READSTRING, WRITESTRING, LENGTH, CONCAT etc.

WIMP – a comprehensive set of 30 routines as provided by ARTHUR.

The routines have been intentionally written to correspond closely to commands in BASIC, relieving the programmer of the need for more manuals, and improving compatibility between languages. The modules have been extensively tested but are in a state of growth as further modules, such as FONTS, are being developed.

The Modules have also been created with a view to encouraging good program design through modularisation. Each implementation has an associated definition file for inclusion into programs to ensure correctness and ease of use (see example below).

ALL of these Modules together with documentation, some sample programs and module source code are available at a cost of £30 from : I.M.Smith, 77 Edward Road, Fleckney, Leicester, LE8 OAD.

Upgrades will be made available when new modules are developed or existing ones extended.

An example of how a MODULE 'Mod.Useful' might be built and used :

```
FILE :- Mod.Useful
MODULE Useful;
{ This contains any of YOUR commonly
used Pascal routines }
EXPORT FUNCTION Cap(Ch : CHAR) :
                                CHAR;
{ Converts a lower case letter to
  upper case but leaves all other @
  characters unchanged }
BEGIN
  IF (Ch IN ['a'..'z']) THEN
    @Cap := CHR(Ord(Ch) - ORD('a') +
                                ORD('A'))
END;
EXPORT PROCEDURE Inc (VAR N :
INTEGER ; Amount : INTEGER);
{ N is increased by Amount }
BEGIN
  N := N + Amount
END;
{ Add other routines here }
END.
FILE :- Inc.Useful
IMPORT FUNCTION Cap(Ch : CHAR) :
                                CHAR;
```

```
{ Converts a lower case letter to
  upper case but leaves all other
  characters unchanged }
IMPORT PROCEDURE Inc (VAR N :
INTEGER ; Amount : INTEGER);
{ N is increased by Amount }
FILE : Pas.Test
PROGRAM TestUsefulRoutines(INPUT,
                                OUTPUT);
$INCLUDE 'Inc.Useful' {INCLUDES the
  source code in Inc.Useful
  given above}
VAR
  Number : INTEGER;
  Letter : CHAR;
BEGIN
  WRITE('Enter a lower case letter
                                ');
  READLN(Letter);
  Letter := Cap(Letter);
  WRITELN('This is now ',Letter);
  WRITE('Enter a Number ');
  READLN(Number);
  Inc(Number,5);
  WRITELN('With 5 added this is
                                ',Number:1);
END.
```

Steps to Compile, Link and Run :

- Compile Mod.Useful to form aof.Useful. Assumes directory Mod for Modules.
- Create Inc.Useful to contain those routines which can be imported.
- Compile Pas.Test to form Aof.Test.
- Link Aof.Test,Aof.Useful,Paslib to form executable file Test.
- Run Test b *Test.

Below in the contents of Inc.General, some generally used 'BBC BASIC' type routines.

```
Inc.General ( The routines
provided)
IMPORT PROCEDURE Vdu(Code :
                                INTEGER);
{ Vdu(7); rings the bell }
IMPORT PROCEDURE Mode(N:INTEGER);
{ Mode(12); Selects MODE 12 }
IMPORT PROCEDURE CIs;
{ Clears current TEXT Screen }
IMPORT PROCEDURE Colour (c :
                                INTEGER);
```

```

{ Colour(1);Colour(132); Set text
  f/g or b/g colours }
IMPORT PROCEDURE
SetPalette(n,p,r,g,b : INTEGER);
{ As VDU 19 in BASIC. If r,g and
  b are 0 then only n and p are as
  in Colour 6,2 which sets actual
  colour 6 to logical colour 2 }
IMPORT PROCEDURE
RestoreDefaultColours; { Self
  explanatory }
IMPORT PROCEDURE
TextWindow(lx,by,rx,ty :
INTEGER);
{ Redefines the text window}
IMPORT PROCEDURE Tab(column,row :
  INTEGER);
{ As in Basic }
IMPORT PROCEDURE Osci (S :
  Strings);
{ Causes ANY * command to be
  executed. e.g. Osci(Cat) or
  Osci(FX 5,2) }
IMPORT PROCEDURE ff;
{ Turns the cursor OFF as in
  BASIC }
IMPORT PROCEDURE ON;
{ Turns the cursor ON as in BASIC }
IMPORT FUNCTION Inkey(T :
  INTEGER) : CHAR;
{ Ch := Inkey(100) as A$ =
  Inkey$(100) in Basic }
IMPORT FUNCTION Get : CHAR;
{ Ch := GET as A$=GET$ in Basic }
IMPORT FUNCTION InkeyNeg (Key :
  INTEGER) : BOOLEAN;
{ REPEAT UNTIL INKEY(-99) in both
  BASIC and Pascal}
IMPORT PROCEDURE GetTime(VAR t
  :strings);
{ Returns the time read from
  internal clock as a string }
IMPORT FUNCTION Time :INTEGER;
{ As in BASIC Now = TIME becomes
  Now := Time }
IMPORT PROCEDURE SetTime(Val
  :INTEGER);
{ As in BASIC TIME = 0 becomes
  SetTime(0) in Pascal }
IMPORT FUNCTION FNhrs :INTEGER;

```

```

{ Returns the Hours component of
  the Time string as an INTEGER }
IMPORT FUNCTION FNmin :INTEGER;
{ Returns the Minutes component
  of the Time string as an
  INTEGER }
IMPORT FUNCTION FNsec :INTEGER;
{ Returns the Seconds component
  of the Time string as an
  INTEGER }

```

On a final note – it seems a great pity that Acorn have decided not to release their in-house MODULA-2 Compiler for the 300 series machines. Wirth's specification for this language extends the ideas of Modular programming and would have been a step towards the future and more in keeping with the Archimedes itself!! (A version of Modula-2 is available – See the advert opposite! Ed.) **A**

Contact Box

- **Anyone doing anything on German?** If so, contact Jim Clarke, Nottingham High School, Waverley Mount, Nottingham, NG7 4ED.
- **Art Teachers.** Steve Bruntlett is doing a lot of work on Artisan and Arctist and would like to get in contact with others. Contact him at 16 Dawson Lane, Bierley, Bradford, BD4 6HN. (Phone 683717, Friday to Tuesday evenings, before 9.00 p.m. if possible.) He's working on a Masters degree about the use of art packages in schools and would be pleased to contact anyone using Artisan in a school or college situation.
- **Scottish Archimedes Users!** There is an existing user-group for Central Scotland which has monthly meetings. For details, contact David Davidson, 2 Akarit Road, Larbert, Stirlingshire, FK5 4BY. (0324-558692)
- **Anyone interested in astronomy?** If so, have you transcribed Peter Duffett-Smith's programs from "Astronomy with your Personal Computer"? Contact D A A Fagandini, 6 Alleyn Park, Dulwich, London SE21 8AE. (Phone: 01-670-0547) **A**

Modula-2 Compiler

A complete interactive environment for developing modula-2 software

Complete debugging environment
Maths and system libraries
Full screen editor
Produces optimized stand alone code
Independent compilation for modules
Full Arthur and WIMP support
Numerous example programs with documentation
Supports hardware floating point processors

Price £89.95

RISCFORTH

Riscforth is a new, 32-bit implementation of the FORTH-83 standard, designed to take full advantage of the ARM Architecture.

Features include:

Multi-tasking	Optimizing Compiler
ARM Assembler	Full Screen Editor
File System Interface	Operating System Calls
Floating Point	Graphics Support
Hashed Dictionary Structure	Single-Step Debugger
Shadow Screen for Documentation	
Utilities for Block Manipulation, Dictionary and Vocabulary Display, Call Finding etc.	
Stand Alone Code Generation with no royalties payable	
Extensive Manual and many example programs	

Price £69.95

Blue Grey Software

28 Bobbitts Way, Wivenhoe, Essex CO7 9NJ

SigmaSheet Review

Rob Brown

"Is it a bird? Is it a plane? No it's SigmaSheet!" That is how one's first impressions of Minerva System's new spreadsheet, SigmaSheet may well be clouded by the advertising. It describes it as both the "fastest" and "largest" as well as being as "powerful" as Lotus 123. The purpose of this review is to try and describe both the features of the software itself as well as briefly describing a practical application to which I have put the product. I should add that this review is based only on two week's use of SigmaSheet and there are therefore some features which I have yet to try out!

First Impressions

The package arrived complete with an A5 ring bound manual (running to nearly 80 pages) as well as the program disc itself. At first sight the manual appears less daunting than that for System Delta Plus but in fact is only some 20 pages shorter. The program disc continues Minerva's policy of providing their software in a 'protected' form. Although it is possible, and actively recommended by Minerva, to take a back-up copy of the program disc for everyday use, it is still necessary to re-insert the original program disc briefly in Drive 0 before loading the main program(s). This also applies if you are using the software on an Archimedes fitted with a hard disc. Minerva are apparently considering other options for their software (e.g. providing ROM versions) and do state that non copy-protected versions are available at extracost!

The true test of any software is how easy is it to use with little or no reference to the documentation. My own experience was to take a cursory glance through the manual followed by the 'both feet first' principle. This approach is probably fine for those users who have had experiences of other spreadsheet type software. Certainly the claim of the "largest" spreadsheet did appear justified with cell references running

from A1 to AM16383, this compared most favourably with my previous experiences of Inter-Sheet on my BBC Micro (A1 to BL255) and even Symphony on the IBM/Compaq PCs at the office (A1 to IV8192).

SigmaSheet takes some advantage of the WIMP environment afforded by the Archimedes, but nowhere near to the same extent as System Delta Plus; furthermore I found having to use the mouse rather difficult to get used to, since for most data entry it is unnecessary. An alternative option is to use <ctrl> in conjunction with appropriate letters rather than using the mouse to 'click' on the various lettered Menu options; certain additional functions are also available by use of <alt> (e.g. <alt-R> performs a manual recalculation). Although the function keys are not used, it would also be perfectly feasible for one to set up the most frequently used options as *KEY definitions.

Documentation

The manual seems to be fairly clearly laid out with sensible, if somewhat sparse, screen illustrations. The contents comprise:

- Introduction
- General Information
- Spreadsheet Environment
- Experimental Section
- Creating a Spreadsheet
- Formulae and Functions
- Command Menu
- Options Menu
- Importing/Exporting Data

followed by sections on Advanced Use, Errors, Quick Reference Guide and an Index. It is worth noting that the section on Errors only details specific error messages generated by SigmaSheet. For a full list one is referred to the User Guide! The novice user is specifically directed to the Experimental Section and is further guided by having two demonstration

files provided on the program disc. However, I did find specific problems with certain elements of the documentation which are noted below.

Capabilities

The program disc is loaded in the usual way by pressing <shift-break>; there is even an additional program which will be loaded if necessary to correctly configure the Archimedes. However, since all of the spreadsheet is held in memory at once, I doubt if A305 users will be able to create very large spreadsheets as the SigmaSheet programs themselves appear to occupy nearly 150k of memory. Upon loading, an initial banner screen is displayed and by pressing the S key one is taken straight into a blank spreadsheet display.

Cells are referenced in the familiar spreadsheet way (e.g. A23), but can also be referenced by user defined labels applying to specific columns and/or rows; one nice touch is that references to areas can be separated by ":" or ".." (the latter being very familiar to users of Lotus products). Cells may contain text, values or formulae and these are supported by a full range of mathematical, statistical and even date functions. Both relative and absolute cell referencing can be used for copying, loading and merging data. Calculations can be set to manual or automatic, and I found the former method necessary with larger spreadsheets as data entry quite noticeably slowed down. A further commendable idea is the "Intelligent calculation" method which automatically takes care of any forward references by calculating the forward referenced cell(s) first.

The Command Menu contains the usual sort of options for: Automatic cursor movement, Copy, Delete, Edit, Format (e.g. display of negative, currency, date/time, decimal values etc. plus a protect option), Go to, Insert, Change mode (80 or 132 column), Load spreadsheet, Options (sub-menu), Print (options/export), Quit, Recalculation mode, Save spreadsheet, Title, Width as well as options for entering *

commands and performing calculations – <escape> is used to return to the main spreadsheet. Where appropriate these options can apply for individual cells, an area, a column, a row, or the whole spreadsheet. There is one apparent omission in that there is no 'move' option, but this can obviously be achieved by way of a 'copy' followed by a 'delete'.

There is a further sub-menu which provides options to control: Colours (for the cell cursor, sheet area, sheet border), Decimal format (., or .), Editing, Formula display, Column gap, Highlight protected cells, Printing and certain user formats – <escape> is again used to return to the main spreadsheet. It was in trying to use the printing options that I first found failings in the documentation. Although two pages are devoted to this subject it was more by trial and error that I was able to produce printed output. I am not sure if this is one of Minerva Systems' particular documentation weaknesses (or one of mine as a user!), but I had previously experienced similar problems with System Delta Plus!

Data Import and Export

Unlike other integrated packages now becoming available for the Archimedes, SigmaSheet is a stand alone spreadsheet and it must therefore be considered essential for appropriate utilities to be available for both importing and exporting data. The DeltaLink utility (selected from the initial SigmaSheet banner screen) provides facilities for importing data from other Minerva products (e.g. System Delta Plus). There is a more general purpose utility (again selected from the banner screen) provided for importing data from other selected products (e.g. Inter-Sheet) plus a general purpose (ASCII format) routine. Exporting of data is achieved via the print (to file) option from the Command Menu, again in ASCII format.

The documentation of this part of the software is very poor indeed, with some serious ambiguities, errors and omissions in the

narrative and with certain described features either missing or different on the actual screen displays. I had to resort to a lengthy telephone call to Minerva for them to put me on the right track and for a further period of experimentation before I was able to achieve a successful data import into SigmaSheet from a System Delta Plus file. It was also at this stage that I found out how limited was the range of error documentation in the manual!

However, the 'external' import/export routines for SigmaSheet appeared to work well, as I was able to import a file from Inter-Sheet and export another file to Inter-Word without any difficulty whatsoever! The documentation on these is limited in the manual but the program(s) incorporate appropriate guidance within the screen display(s).

SigmaSheet in Action

As treasurer for a non-incorporated body, I have been instrumental in computerising the records both in terms of name/address data etc (using System Delta Plus) and also dealing with the accounting of same. The latter was previously handled using Inter-Sheet on my BBC Micro but with an increasing volume of transactions, several files had to be used to hold the whole financial year's data. A further complication is that a bank account was opened at the start of the financial year for our member organisations in Ireland, consolidation of foreign currency (IR£) thus became necessary.

The spreadsheet itself consists of a number of elements of data:

- Income (including detailed analysis of each receipt)

- Expenditure (including detailed analysis of each payment)

- Cash book summary

- Profit & Loss (including comparative figures)

- Balance Sheet (including comparative figures)

with the above being repeated for IR£ transactions, there are then further summaries:

- Currency conversion (IR£ to UK£)
- Consolidated Profit & Loss Account
- Consolidated Balance Sheet

One of the reasons for using SigmaSheet for the accounting function is that a spreadsheet does afford the flexibility that a 'package' solution may lack; cost is another consideration.

Having received SigmaSheet in the post on a Saturday morning, I went through a rapid read of the manual before both setting up the above comprehensive spreadsheet and inputting of data for the whole of the financial year to date by the end of the recent Bank Holiday weekend. We also had other commitments that weekend, so it is clear that the learning curve for SigmaSheet is fairly rapid. However, I had not at that stage attempted any form of data import or export.

Conclusion

To the question "is it worth buying", the answer must be "yes" for someone who wants the capacity and facilities of equivalent PC based spreadsheet packages at a fraction of the price (i.e. £69.95). The question "does it live up to its expectations", I feel that the answer must be that it is an otherwise excellent product let down by some shortcomings in its documentation. **A**

Comments from Other Readers

The Editor has received one or two comments from other readers which may add to the overall view of the product. One reader sent us a list of things which he felt were wrong with SigmaSheet but ended up by saying that Minerva were the **first** and **only** company he had come across who were prepared to listen, consider and implement changes suggested by a consumer. A similar comment came from someone who spoke to us at the Show.

On the negative side, there have been some unfavourable comparisons, in terms of facilities, with Inter-Sheet. Firstly, when specifying box numbers for copying (or entering formulae), the sheet itself is fixed so you have to decide the box numbers you want, memorise them or write

Logistix – Spreadsheet Plus?

Mark Sealey

Logistix is another in the series of major pieces of software which alone almost justify the purchase of an Archimedes. Originally written for the IBM PC range (and clones) and available for some time on the Atari ST series, it is an extremely complex and powerful suite that will appeal, as they say, to the serious user.

What is Logistix?

At its heart is a spreadsheet with just about as many features as anyone could want. If that were all, the price Acorn is asking would seem a little high. But onto the basic spreadsheet idea have been grafted – with total success – time and job management facilities that anyone who believes in planning an activity will delight in.

Add to this an extremely flexible set of graphics options and some measure of critical path analysis of tasks as well as a programming language and database and you have a very

them down, then call /C and type them in. This is very 'wooden' compared with Inter-Sheet where you can call /C then move the cursor around the sheet and either type in the box numbers or, better still, put the cursor over the required box and press <copy> which types in the box number for you – no need to look up and down to work out the number, and much less chance of getting the wrong box number and maybe wiping out valuable information.

Secondly, we have not found any way of 'holding' a line on the screen. If you have a great long list of figures, how do you remember which column is which? On Inter-Sheet you can /Hold a row (or column) at a particular place on the screen. The sheet then scrolls around underneath it, so that the titles are always visible on screen. Similarly, if you want to run your eye down two columns it's very difficult if they are too far apart (even in 132 columns) to see on-screen at the same time. Even if they are both visible on

impressive piece of software indeed. It is a real advantage that each section of the package does use the spreadsheet format, rather in the way that Viewstore has a spreadsheet as well as a card-index display.

The Learning Curve

With something as flexible and comprehensive as this suite, however, the greatest danger is always that the control structure is so dense that using it only becomes possible when you have spent hours studying a manual and even longer learning from ill-understood mistakes.

Not a bit of it here. Logistix is not menu driven as such, but relies on single-letter commands prefaced usually by a "slash" (/) yet where only those applicable within any given context are allowed. For example, to insert a row you can either type '/' then 'I' then 'R' or '/IR' without waiting. Escape cancels the current command – always – and after a few minutes, you feel completely in control. It really is one of those

screen, it's so much easier to compare two columns when they are side by side. Is there any way round this with SigmaSheet?

Thirdly, users of earlier versions of SigmaSheet will find that the Import from ViewSheet function is unusable for all but a very small sheet. This bug has now been corrected.

Finally, there is no request for confirmation when you are loading and saving sheets – even when doing so would over-write a sheet either in memory or on disc. Another simple thing which could easily have been incorporated but was just not thought of at the time.

*If you have views on the relative merits of any of the Archimedes software, please send it in – not just brickbats, please, because we need to know positively which packages are best for different applications and it's you, the user, who can best advise other users which to buy because you have tried it and have no axe to grind. **A***

pieces of software that you can pick up and use practically straightaway.

You can set up a new spreadsheet in minutes and toggle a whole set of options relating to entry format, recalculation (which is very quick) and the like. Pressing <F1> will always bring you appropriate and clear help screens relating to the activity currently engaged in. A status line also carries messages for all operations. Although you can alter them, the colours that Logistix uses as default add to the extreme user-friendliness; the mouse and WIMP environment are not used at all but this is really an advantage where data entry is concerned.

The Worksheet

The basic concept of Logistix is the worksheet. Indeed the longest chapter in the excellent (but weighty, two volume) manual is devoted to getting the most from the core environment. It is within this that all slash commands originate. But where data is concerned, the idea of combining more than one use is more like View Professional than, say, Multiplan or Ultracalc. Given that it is possible, then, to have a job analysis area, a database and a conventional spreadsheet all occupying different parts of your 'worksheet' at the same time, learning the system does not take at all long; you scroll around (with or without the use of windows) and make additions or adjustments as you need.

There are many statistical and mathematical functions together with Calendar and Boolean Logical operations to be used at most stages. Purely financial functions include ones on interest, mortgage (i.e. principal, rate, term) and values. The flexibility of display and control of the graphical representations of data are huge – with a whole array of axis-scaling and labelling options, all excellently explained in the manuals. Indeed the programming and job-handling facilities alone could fill a series of specialist tutorials for prospective users.

Suppose you were the classic sales rep who has to visit several cities and towns without covering

the same ground twice and yet stocking her/himself to fulfil all orders in the correct sequence. Let's say that you have always been inhibited by the complexity of the variables of such a task; well, Logistix is ideal and can help both to understand and display the results not only of the basic situation, but of various hypotheses relating to it admirably.

Furthermore, where it gets exciting is in linking your data to the calendar and watching, for example, as the outcome of a change in the time needed for your broad beans to sprout because of a wet winter alters the starting date for picking them – dynamically on the screen as happens conventionally with financial calculations. Quite sophisticated automated and animated presentations can be handled with real ease in this way, thanks to the inbuilt language à la 'dBase II' and 'III+'. There are several sample files on the disk too.

There is an impressive battery of printer options and I had no difficulty customising it to my own model. Although it will give 8 pin dumps of all the graphics and text in a variety of fonts and text-styles, 24 pin configurations are not catered for. Remember to have Logistix files temporarily set to Read/Write and unlocked as this and other customisation data is saved not to a separate file but to one of the system files on the disk. What is more, some versions of the program appear not to allow changing of disks and subsequent remounting – as when you maintain a separate data disk in Drive 0. The version I tested worked perfectly with two internal drives, though it did assume data was on Drive 1. This is maybe a throwback to the suite's non-Acorn origins. I have also used a version of Logistix under MS-DOS, which, although running more slowly than this Archimedes version, more happily accommodates some of the help pages that refer to files called "budget.doc" and "A:\TIME.LGX"!

All in all, Logistix is well worth buying. Sigma-Sheet from Minerva arrived just too late to be

Using the Debugger – BreakPoints

Gerald Fitton


Break Points

When a machine code program is CALLED it usually (or rather, hopefully) runs without a hitch until it returns to BASIC in an orderly way with the command "MOV PC,link". A break point is an address, specified by the programmer, where the execution of the program is stopped and control returned to the user. Acorn's debugger makes provision for up to 16 break points (addresses) to be set by the programmer. To set a break point the debugger replaces the instruction at the break point address with an unconditional branch (i.e. Branch ALWAYS) to the debugger code. This debugger code saves the values held in the 16 ARM registers into the debugger's workspace and allows the user to enter any "*" command. One such command is "*Continue": this command replaces all the saved registers, including the value of the program counter (R15=PC) and so the program continues to execute from the break point address.

Directories

I have been reminded that hard discs are no longer the luxury that they once were. To use a hard disc effectively, the "\$" directory should contain almost nothing but directories. Even if you intend to use floppies for ever then at least you should learn about "hierarchical directories". I still don't have my hard disc but here is the method I use as applied to this set of programs.

looked at in depth and, despite probably being faster and of greater capacity, lacks completely the time and job management facilities of Logistix. The latter make it a comprehensive and very useful tool for anyone handling any sort of live, flexible, human' data in quantity with a lot of "what if" operations to perform reliably and quickly.

Logistix from Acorn Computers, price £99 + VAT. (£105 inc. VAT through Archive.) 

First create a directory with "*CDIR Debugger" and make it your User directory with "*URD Debugger". This has the effect that "&" in file pathnames will always be interpreted as the "Debugger" directory. Create "BasicProgs", "ObjectCode" and "SourceCode" as sub-directories of the "Debugger" directory. When the program "&.SourceCode.Alphabet01" is run, it will save a file into the "&.ObjectCode" directory called "Alphabet01".

The Program Alphabet01

To demonstrate the break point commands of the debugger we need some code. The 50 line program "&.SourceCode.Alphabet01" is a self-contained code generating function which can be used as a stand-alone program or it can be called as a LIBRARY function with "Memory02". It prints on the screen the capital letters of the alphabet through the loop of lines 30370 to 30410. Before you run the program type SAVE (from anywhere in the directory structure) and, provided your URD is "Debugger" and, if the "Debugger.SourceCode" sub-directory exists, then the program "Alphabet01" will be saved in "Debugger.SourceCode". Run the program as a stand alone program (again from anywhere on the disc) and the executable object code file, "Alphabet01" will be saved (line 30140) into "Debugger.ObjectCode". Correct any typing errors now before running the next program. Bugs in LIBRARY routines give incomprehensible error messages.

The Program BreakPts01

This 50 line program sets up seven function keys with *-commands from the debugger at lines 210 to 270. The self-contained code generating function "Alphabet01" is loaded into the LIBRARY at line 280 and the code assembled through line 290. Lines 180 and 190 guarantee that the code will be assembled at &10000,

which is useful since it matches "Memory02". A single break point is set by the program (line 310) at address &10000, the start of the machine code program. Other break points can be set by the user of the program whilst it is running. Type in the program and SAVE it.

Using the BreakPts01 Program (1)

When you run "&.BasicProgs.BreakPts01" the program "&.SourceCode.Alphabet01" is loaded into memory by line 280 and the code contained within it is assembled in memory at &10000 by line 290. The assembler has a screen display generated by the value of pass% chosen in line 30330 of Alphabet01. At the end of the display, the message "Press any key to continue" is generated by line 320. Before continuing further press <escape> followed by "PRINT ~!&10000". My machine gives EA5FC137 as the hex. code contained by &10000. This is not the code we put there as you will see! Now press the red key <f7> followed by <return> and you will see that it gives the contents of &10000 as E92D4000. This is the code we put there. Press <escape> and run the program again. "Press a key to continue" and you will find that at line 350 the code is disassembled using *MemoryI. Look at the display for &10000 and you will see a "*" where other locations have ":". This implies that the debugger has set a breakpoint at that address. What has happened is that the original contents, E92D4000, have been replaced by EA5FC137 which has a different effect. Before continuing further press <ctrl-reset> to reset the debugger workspace and then type "*MemoryI 10000 +30" (note the space between 10000 and the + sign). You will find that the disassembled version of EA5FC137 at location &10000 is an unconditional branch to &018004E4, which is within the debugger workspace.

Using the BreakPts01 Program (2)

Type OLD <return> to recover your program (and *URD \$.Debugger), run it again pressing "any key" when invited to continue. You will halt within the CALLED code (line 390) as a

result of the unconditional branch at &10000, the first breakpoint. The values held in the 16 ARM registers are transferred to the debugger workspace (on my machine, to &01800584) and then displayed. The registers do not actually contain these values when you see them. The display is what would be in the registers just before executing the (original) instruction. In the User Guide under the command "CALL", you will find that registers 8 to 14 contain information about the BASIC environment that you have just left. Register 12, for example, points to the current line in the BASIC program. The BASIC command CALL causes the OS to keep a copy of all this information. When you exit from the machine code routine you have CALLED, this information is used to return you to the BASIC program, at the line just after the CALL where you left it.

If you are following the "lesson" described above you should have a flashing cursor just after the prompt "DEBUG*" above which is the "Register dump" preceded by the message "Stopped at break point set at 00010000". Press the red key <f1> (to continue) and <return>. The question "Execute out of line?" is asked. If the breakpoint contains code such as "SUB R3,PC,#&30", which is stored at &10028, then pressing <y> will result in an error because PC, the program counter, contains an address somewhere in debugger workspace. We'll test this later. For now press <y>. The code will execute, printing the alphabet and will return to BASIC in an orderly way halting at line 410.

Press <q> and type in "*MemoryI <register dump> +40" (replacing <register dump> by the register dump location for your machine, 1800584 on mine) and you will see that the registers are those saved at the &10000 break point. Run the program again up to the &10000 break point. Now press <f2> <return> and you will see that you have only one break point. Press <f4> and then type "1002C" <return> followed by <f2>. The BreakList now contains two break points, one at the end of the program. Continue

(using <f1>) as before and you will halt at the new break point 1002C. By pressing <f6> <return> you will get a copy of the disassembled code on screen. R1 contains the value, &5B, MOVED into it by the code at &1000C. R2 contains a copy of R14 (the "link" back to BASIC) and R3 contains the value of code% (&10000). You can now continue (with <f1>) using <y> to continue out of line. We can use <y> even though the mnemonic has a PC in it because in this case we are putting a value into PC from the stack (the value of the "link" stored by the code at &10000) and not loading PC from the debugger workspace.

Using the BreakPts01 Program (3)

Press any key (but not <q>) to run the code again. This time set (with <f4>) a break point at &10028. When asked at break point &10028 "Execute out of line?" press <y> as usual. The code now halts at &1002C. The Register dump does not show "R3 = 00010000" (the "correct" value) but (on my machine) "R3 = 01800660" which is in the debugger workspace. This is the reason why you must not press <y> when the value of PC is MOVED, ADDED or SUBTRACTED from another register. Run the program again up to the &10028 break point. You can experiment with <f1> followed by <n>: you have done nothing. In the book it says that you can clear a break point by BreakClr, <f3>, followed by 10028 <return>. This action restores the original contents of &10028. If you press <f6> you will see that the "*" has disappeared from address &10028. Pressing <f2>, BreakList, confirms that &10028 is no longer a break point. However, in spite of what it says in the book, there seems to be a bug (or is it a feature?) with BreakList which might have shown up when you pressed <f2>. If a single break point is cleared with BreakClr then, when BreakList is called an address exception error occurs. My solution is to look for break points with *MemoryI (key <f6>) rather than BreakList. BreakList does not generate an error if you clear all break points (<f2> <return>) and then

BreakSet the ones you want back. If you use this second technique then, to clear a single break point you BreakClr everything <f3>, and then BreakSet <f4> the ones you want to keep. Anyway, run the program again with break points at &10000 and &10028 only and you will find that when the program halts at &1002C, R3 contains 00010000.

Now for a valid use of *MemoryA. Press <f7> and by skillful(!) use of <delete> change the command to read "*MemoryA 1000C E3A01050". Press <return> followed by <f6> (MemoryI) and you will see that the code at &1000C has been changed. Continuing with the program, results in the letters A to O being printed instead of A to Z. You can use *MemoryA to change the values in the Register dump (including the PC, R15) and then use *Continue which will run the program from the value held in the register dump value of PC. At any time you can *ShowRegs <f5>, and change the values before continuing. I suggest that you have a go with this short program, where you can't do any real harm, before embarking on a more serious project.

Conclusion

Apart from the *BreakList "feature", the Acorn debugger is a useful and bug free tool. It would be a useful addition to be able to enter a mnemonic into a specific memory location from the "DEBUG*" prompt. The program of this article, "&.BasicProgs.BreakPts01", shows most of its features. Some feedback from Archive users would be most welcome.

```

100 REM > &.BasicProgs.BreakPts01
110 REM Author : G L Fitton
120 REM Copyright : ABACUS TRAINING
130 REM Version 1.00 : 1st May 1988
140 :
150 REM A program to demonstrate
      various Debugger commands.
160 :
170 MODE 3
180 DIM dummy% (&10000-TOP-&24)
190 DIM code% &200
200 :
```

Using the Debugger

```

210 *KEY 1 *Continue
220 *KEY 2 *BreakList
230 *KEY 3 *BreakClr
240 *KEY 4*BreakSet
250 *KEY 5 *ShowRegs
260 *KEY 6 *MemoryI 10000 +30
270 *KEY 7 *MemoryA 10000
280 LIBRARY "&.SourceCode.Alphabet01"
290 endofcode%=FN_Alphabet01(code%)
300 :
310 OSCLI("BreakSet "+STR$~(code%))
320 PRINT '"Press any key to
                                continue."';
330 wait$=GET$
340 PRINT '
350 *MemoryI 10000 +30
360 PRINT
370 :
380 REPEAT
390 CALL code%
400 PRINT '"Press Q to quit or any
    other key to CALL the code again."';
410 yesno$=GET$
420 PRINT '
430 UNTIL INSTR("Qq",yesno$)
440 :
450 PRINT "Programme finished."
460 END

30000 REM > &.SourceCode.Alphabet01
30010 REM Author : G L Fitton
30020 REM Copyright:ABABCUS TRAINING
30030 REM Version 1.00:1st May 1988
30040 :
30050 REM The source code for a
    simple machine code program
30060 REM which systematically
    prints the alphabet.
30070 REM Can be used to demonstrate
    the BreakPts01 program
30080 :
30090 MODE 0
30100 DIM dummy% (&10000-TOP-&24)
    :REM A dummy byte array.
30110 DIM code% &200:REM Place code%
    at &10000.
30120 :
30130 endofcode%=FN_Alphabet01(code%)
    :REM Assemble the code.
30140 OSCLI("SAVE &.ObjectCode.
    Alphabet01 "+STR$~(code%)
    +" "+ STR$~(endofcode%))

30150 *SETTYPE &.ObjectCode.
                                Alphabet01 &FFC
30160 *STAMP &.ObjectCode.Alphabet01
30170 :
30180 PRINT'
30190 *MemoryI 10000 +30
30200 :
30210 END
30220 :
30230 DEF FN_Alphabet01(start%)
30240 REM Prints the upper case
                                alphabet.
30250 LOCAL sp,pass%,print
30260 :
30270 REM Use the BASIC stack.
30280 sp = 13: REM Stack pointer.
30290 :
30300 FOR pass% = 0 TO 3 STEP 3
30310 P%=start%
30320 [OPT pass%
30330 STMFD (sp)!,{R14} ;Store
                                link on the stack.
30340 MOV R2,R14 ;Store link in R2
30350 MOV R0,#&41 ; The ASCII code
                                for "A".
30360 MOV R1,#&5B ; The character
                                after "Z".
30370 .print ; A label for BMI
                                print
30380 SWI "XOS_WriteC" ; Print
                                character in R0.
30390 ADD R0,R0,#1 ; Next
                                character into R0.
30400 CMP R0,R1 ; Last character?
30410 BMI print ; If not then
                                goto print.
30420 SWI "XOS_NewLine" ; <LF>,<CR>
30430 SWI "XOS_NewLine" ; <LF>,<CR>
30440 ADR R3,start% ; Address of
                                start% to R3.
30450 LDMFD (sp)!,{PC} ; Return
                                to BASIC prog.
30460 ]
30470 NEXT pass%
30480 =P% A

```

Editing the 256-Colour Palette

Adrian Look

To define a logical colour into any one of the 4096 physical colours, the User Guide tells us to use the colour command:

COLOUR x,r,g,b

where x = logical number

r = red element (times 16)

g = green element (times 16)

b = blue element (times 16)

However, this doesn't work for 256 colour modes. What's more, neither the User Guide nor Programmer's Reference Manual has any information on the subject. We are just advised that it has something to do with the video controller chip (VIDC).

The problem with the 256 colour modes is that they have 256 colours(!) whilst the VIDC chip can only store 16 colours in its memory. So to get 256 colours Acorn have had to resort to trickery. As a result, flexibility had to be sacrificed.

Normally, we can define a colour in terms of its red, green and blue elements – ranging from (0-15) for each of these. This means that a palette definition consists of three, four-bit chunks assigned to each RGB element.

```
1111 1111 1111
red  green  blue
```

In order to get 256 colours from the VIDC chip, the OS takes control of four of these bits (1 red, 2 green, and 1 blue) giving 16 base colours. Allowing us to control the remaining eight bits (3 red, 2 green, and 3 blue), which give 256 possible offset colours.

```
c u u u c c u u c u u
red  green  blue
```

The bits marked c are controlled by the computer, the other bits (marked u) can be defined by the user. Since the computer generates 16 base colours, it is left for us to select a further 16 offset colours from the possible 256. These 16 offset

colours can then be held in the VIDC's memory. The bits from the 16 offset colours are added to each of the 16 base colours giving (16x16) 256 colours. We can define these offset colours by using the COLOUR command for 'logical colours' 0 to 15 – remember only the bits marked with a 'u' will be effective.

Since we will be redefining the palette, the COLOUR and TINT relationships will no longer hold. This means that we will have to forget the present 'colour-tint' system and use a 'base-offset' system. However, BASIC V will still access the colour using the COLOUR and TINT commands. So we must write a procedure which will convert our new system into the colour-tint system. This done at line 2640, i.e.:

```
DEFPROCcolour(base,offset,background)
```

where **base** is the base colour (0 – 15), **offset** is the offset colour (0 – 15), **background** is either **TRUE** to select background colour or **FALSE** to select foreground colour.

Using the Editor

The editor is fully mouse driven and all the functions are effected by the <select> button. The screen is divided into four regions. There is the colour grid, the offset RGB bar display, the full RGB bar display, and the command list (see diagram overleaf).

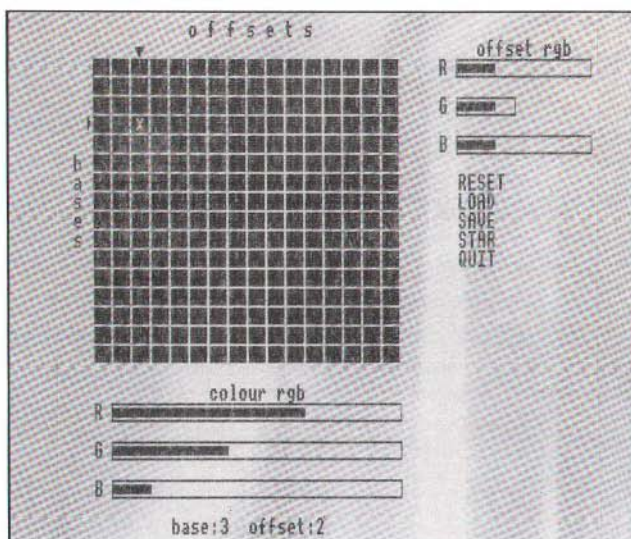
In order to edit a particular colour, you should click on it in the colour grid. Two arrows and a little 'x' will indicate which colour is presently being edited. The full RGB is displayed below the grid. This contains both the offset and base elements combined – this may not be edited.

To the left of the grid, the offset RGB bars are displayed. These are adjusted by positioning the pointer over one of the RGB bars and holding <select> down – the bar can then be increased or decreased as you move the pointer right or left. This will then edit the offset palette definition. Thus not only will the colour selected be

affected but all the other colours that use that offset – just try it and see for yourself!

Finally, you can select any command by clicking on it. The computer will then ask you to confirm your choice – just click on either [Y] or [N]. The RESET command sets all the palette definitions to their defaults. All the other commands should be self explanatory.

The files that are produced by the editor are of the *PRINT format. This means that once you have defined your palette and saved it on disk, you can use the definition in other programs by *PRINTing the file produced by the editor.



```

10 REM >$.256col
20
30 REM *****
40 REM * Setting the Palette *
50 REM * in 256 colour Modes *
60 REM * Adrian Philip Look *
70 REM * 1st March 1988 *
80 REM *****
90
100 MODE 15:OFF
110 *FX 14,6
120 PROCinitialise
130 PROCdisplay
140 *FX 13,6
150
160 ON ERROR PROCerror
170 finished=FALSE
180 REPEAT
190   PROCedit
200 UNTIL finished
210 MODE 0
220 PRINT"Program finished."
230 *FX 15
240 END
250
260 DEFPROCerror
270 IF ERR=17 THEN ENDPROC
280 MODE 0
290 PRINT'REPORT$;" at line ";ERL
300 END
310

```

```

320 DEFPROCinitialise
330 PROCuser_graphics
340 px=0:py=0
350 old_px=-1:old_py=-1
360 command=-1
370 old_command=-1
380 ENDPROC
390
400 DEFPROCuser_graphics
410 VDU 23,255,&7F,&7F,&7F,&7F,&7F,&7F,&7F,&7F,&00
420 VDU 23,254,&FE,&FE,&FE,&FE,&FE,&FE,&FE,&FE,&00
430 VDU 23,253,&00,&00,&00,&00,&0F,&07,&03,&01
440 VDU 23,252,&00,&00,&00,&00,&F0,&E0,&C0,&80
450 VDU 23,251,&00,&08,&0C,&0E,&0E,&0C,&08,&00
460 VDU 23,250,&01,&01,&01,&01,&01,&01,&01,&01
470 VDU 23,249,&80,&80,&80,&80,&80,&80,&80,&80
480 VDU 23,248,&FF,&00,&FF,&FF,&FF,&FF,&FF,&00,&FF
490 VDU 23,247,&FF,&00,&00,&00,&00,&00,&00,&00,&FF
500 ENDPROC
510
520 DEFPROCdisplay

```

```

530 PRINTTAB(2,0);"Archive 256-
      Colour Palette Editor"
540 PRINTTAB(12,2);"o f f s e t s"
550 PRINTTAB(0,9);"b";TAB(0,10);
      "a";TAB(0,11);"s";TAB(0,12);
      "e";TAB(0,13);"s"
560 PRINTTAB(2,4);
570 FOR base=0 TO 15
580   FOR offset=0 TO 15
590     PROCcolour(base,offset,0)
600     VDU 255,254
610   NEXT offset
620   VDU 10,13,9,9
630 NEXT base
640
650 PRINT'TAB(14);"colour rgb"
660 PRINTSPC(1);" R";VDU250:PRINT
      STRING$(30,CHR$(247));VDU249
670 PRINT' 'SPC(1);" G";VDU250:PRINT
      STRING$(30,CHR$(247));VDU249
680 PRINT' 'SPC(1);" B";VDU250:PRINT
      STRING$(30,CHR$(247));VDU249
690 PRINT' ' ' ' ' ' written by Adrian
      Philip Look"
700
710 PRINTTAB(42,3);"offset rgb"
720 PRINTTAB(37,4);" R";VDU250:PRINT
      STRING$(14,CHR$(247));VDU249
730 PRINTTAB(37,6);" G";VDU250:PRINT
      STRING$(6,CHR$(247));VDU249
740 PRINTTAB(37,8);" B";VDU250:PRINT
      STRING$(14,CHR$(247));VDU249
750
760 PRINTTAB(40,10);"RESET"
770 PRINTTAB(40,11);"LOAD"
780 PRINTTAB(40,12);"SAVE"
790 PRINTTAB(40,13);"STAR"
800 PRINTTAB(40,14);"QUIT"
810
820 PROCupdate
830 *POINTER 1
840 ENDPROC
850
860 DEFPROCupdate
870 GCOL 3,63 TINT 192
880 VDU 5
890 MOVE (px+1)*32+8,(32-(py+4))*
      32:PRINT"x"
900 MOVE (px+1)*32,928:VDU 253,252
910 MOVE 16,(28-py)*32:VDU 251
920 VDU 4:OFF
930 PROCrgb(15,15,15,4,22)
940 PROCrgb(7,3,7,40,4)
950 IF old_px=px AND old_py=py THEN
      ENDPROC
960 old_px=px:old_py=py
970 COLOUR 63 TINT 192
980 PRINTTAB(4,28);SPC(39);
990 PRINTTAB(10,28);"base:";py;
      TAB(18,28);"offset:";px
1000 ENDPROC
1010
1020 DEFPROCrgb(mr,mg,mb,rgbx,rgby)
1030 SYS "OS_ReadPalette",px,16 TO
      d,d,c
1040 r=(c DIV &1000) AND mr
1050 g=(c DIV &100000) AND mg
1060 b=(c DIV &10000000) AND mb
1070 IF mr=15 THEN
1080   gcol=px+py*16
1090   IF (gcol AND &10)>0 THEN r+=8
1100   IF (gcol AND &40)>0 THEN g+=8
1110   IF (gcol AND &20)>0 THEN g+=4
1120   IF (gcol AND &80)>0 THEN b+=8
1130 ENDIF
1140 PRINTTAB(rgbx,rgby);STRING$
      (r*2,CHR$(248));STRING$
      ((mr-r)*2,CHR$(247))
1150 PRINTTAB(rgbx,rgby+2);STRING$
      (g*2,CHR$(248));STRING$
      ((mg-g)*2,CHR$(247))
1160 PRINTTAB(rgbx,rgby+4);STRING$
      (b*2,CHR$(248));STRING$
      ((mb-b)*2,CHR$(247))
1170 ENDPROC
1180
1190 DEFPROCedit
1200 MOUSE x,y,buttons
1210 dx=x DIV 32:dy=31-y DIV 32
1220 IF dx>18 AND dx<23 AND dy>9 AND
      dy<15 THEN command=dy
      ELSE command=0
1230 IF command<>old_command THEN
      PROCselect
1240 IF buttons<>4 THEN ENDPROC
1250 IF dx>0 AND dx<17 AND dy>3 AND
      dy<20 THEN PROCnew_colour
1260 IF x/32<19.5 THEN ENDPROC
1270 IF command<>0 THEN
1280   CASE command OF
1290     WHEN 10 : PROCreset
1300     WHEN 11 : PROCload
1310     WHEN 12 : PROCsave
1320     WHEN 13 : PROCstar

```

```

1330     WHEN 14 : PROCquit
1340     ENDCASE
1350 ENDIF
1360 IF dy=4 AND dx<27 THEN
      PROCset_palette(dx,-1,-1)
1370 IF dy=6 AND dx<23 THEN
      PROCset_palette(-1,dx,-1)
1380 IF dy=8 AND dx<27 THEN
      PROCset_palette(-1,-1,dx)
1390 ENDPROC
1400
1410 DEFPROCreset
1420 PROCare_you_sure
1430 IF answer=1 THEN VDU 20
1440 ENDPROC
1450
1460 DEFPROCload
1470 PROCare_you_sure
1480 *fx 14,6
1490 IF answer=2 THEN ENDPROC
1500 PROCfilename:IF filename$=""
      THEN ENDPROC
1510 temp=px
1520 LOCAL ERROR
1530 ON ERROR LOCAL PROCerror1
      :ENDPROC
1540 *MOUNT
1550 OSCLI("PRINT "+filename$)
1560 PROCrgb(7,3,7,40,4)
1570 *FX 13,6
1580 ENDPROC
1590
1600 DEFPROCsave
1610 PROCare_you_sure
1620 *fx 14,6
1630 IF answer=2 THEN ENDPROC
1640 PROCfilename:IF filename$=""
      THEN ENDPROC
1650 temp=px
1660 LOCAL ERROR
1670 ON ERROR LOCAL PROCerror1
      :ENDPROC
1680 *MOUNT
1690 file=OPENOUT(filename$)
1700 FOR px=0 TO 15
1710   PROCrgb(7,3,7,40,4)
1720   BPUT #file,19
1730   BPUT #file,px
1740   BPUT #file,16
1750   BPUT #file,r*16
1760   BPUT #file,g*16
1770   BPUT #file,b*16
1780 NEXT px
1790 px=temp
1800 CLOSE #0
1810 PROCrgb(7,3,7,40,4)
1820 *fx 13,6
1830 ENDPROC
1840
1850 DEFPROCfilename
1860 *POINTER 0
1870 ON
1880 INPUTTAB(40,16)"Filename">
      filename$
1890 PRINTTAB(40,16);SPC(39);
1900 OFF
1910 *POINTER 1
1920 ENDPROC
1930
1940 DEFPROCerror1
1950 VDU 7
1960 PRINTTAB(40,16);REPORT$
1970 TIME=0:REPEAT:UNTIL TIME>100
1980 PRINTTAB(40,16);SPC(39);
1990 px=temp:PROCrgb(7,3,7,40,4)
2000 *fx 13,6
2010 ENDPROC
2020
2030 DEFPROCstar
2040 PROCare_you_sure
2050 IF answer=2 THEN ENDPROC
2060 *POINTER 0
2070 VDU 28,40,31,79,16:ON
2080 LOCAL ERROR
2090 REPEAT
2100   ON ERROR LOCAL REPORT:PRINT
2110   INPUT"*"input$
2120   OSCLI(input$)
2130 UNTIL input$=""
2140 VDU 12,26:OFF
2150 *POINTER 1
2160 ENDPROC
2170
2180 DEFPROCquit
2190 PROCare_you_sure
2200 IF answer=1 THEN finished=TRUE
2210 ENDPROC
2220
2230 DEFPROCare_you_sure
2240 *FX 14,6
2250 VDU 7
2260 PRINTTAB(40,16);"Are you sure?"
      ;TAB(40,17);"[Y] or [N]"
2270 answer=-1
2280 REPEAT

```

Screen Conversions

Gwyneth Pettit & Mike Williams

Converting BBC Screens

Gwyneth Pettit (Academic Software)

Because of the extra graphics facilities of the Archimedes, it will be rare for screendumps to be carried over to the Archimedes from earlier Acorn micros. We had a particular need to convert our 'house style' logo screendump from the BBC micro, and also some diagrams for simulations, which we would have found hard to 'redraw' from scratch on the Archimedes. So although it is not a priority for most Archimedes users, there may be others in Archive who need the same facility, to whom we offer this program in the interests of saving programmer power.

First you must transfer your screendump to an Archimedes disk – we achieved this by the dual drive attached to our Master which has one 5.25" drive and one 3.5" drive. You may prefer to move it via the RS423 or you may have a 5.25" disk on your Archimedes. Either way, you must have the original screendump in ADFS format on your Archimedes disk before you can run the conversion program.

As it stands, the program will write the converted screendump back to the disk which holds the original screendump, and will append the characters ARC to the input filename. This preserves the original dump while making the new name instantly recognisable, but since

```
2290  MOUSE x,y,buttons
2300  dx=x DIV 16:dy=y DIV 32
2310  IF dy=14 AND buttons=4 THEN
2320      IF dx=41 THEN answer=1
2330      IF dx=48 THEN answer=2
2340  ENDIF
2350  UNTIL NOT answer
2360  PRINTTAB(40,16);SPC(20);
      TAB(40,17);SPC(20)
2370  *FX 13,6
2380  ENDPROC
2390
2400  DEFPROCset_palette(nr,ng,nb)
2410  IF nr<>-1 THEN r=dx-19
2420  IF ng<>-1 THEN g=dx-19
2430  IF nb<>-1 THEN b=dx-19
2440  COLOUR px,r*16,g*16,b*16
2450  PROCrgb(15,15,15,4,22)
2460  PROCrgb(7,3,7,40,4)
2470  ENDPROC
2480
2490  DEFPROCnew_colour
2500  PROCupdate
2510  px=dx-1:py=dy-4
2520  PROCupdate
2530  ENDPROC
2540
2550  DEFPROCselect
2560  PRINTTAB(38,10);" ";TAB(38,11);
      " ";TAB(38,12);" "
2570  PRINTTAB(38,13);" ";TAB(38,14);
      " ";TAB(38,15);" "
2580  PRINTTAB(46,10);" ";TAB(46,11);
      " ";TAB(46,12);" "
2590  PRINTTAB(46,13);" ";TAB(46,14);
      " ";TAB(46,15);" "
2600  IF command<>0 THEN PRINTTAB
      (38,dy);">";TAB(46,dy);"<"
2610  old_command=command
2620  ENDPROC
2630
2640  DEFPROCcolour(base,offset
      ,background)
2650  n=base*16+offset
2660  rb1=((n AND &04)>0)
2670  rb2=((n AND &10)>0)*2
2680  gb1=((n AND &20)>0)*4
2690  gb2=((n AND &40)>0)*8
2700  bb1=((n AND &08)>0)*16
2710  bb2=((n AND &80)>0)*32
2720  white=n AND 3
2730
2740  tint=white*64
2750  colour=rb1+rb2+gb1+gb2+bb1+bb2
2760  IF background THEN colour+=128
2770
2780  COLOUR colour TINT tint
2790  ENDPROC A
```

ADFS names are limited to 10 characters, it also means you must keep the original name to the seven characters allowed under DFS. If you wish to overwrite the original screendump, change line 410 to read

```
410 newfile$=file$+" "
```

since the space is essential for the OSCLI command in line 140. If you are working on a 440, you can call the original screendump from the Winchester by typing :4.<filepath>, or from the floppy disk by typing :0.<filepath> when prompted (see line 370) by

Give name of BBC screen dump to be converted ?

You will also be prompted to enter the original screen mode (see line 250) and the converted screen dump will be in the same mode for the Archimedes. Note that only screen dumps from the graphics modes (0, 1, 2, 4 and 5) can be converted by this program; if you dumped from modes 3, 6 or 7 you have to solve the problem your own way. Most of the screen information for the Archimedes dumps was discovered by the time-honoured method of heuristics, since there is no Archimedes equivalent to the BBC Advanced User Guide (yet!).

The Screen Convert program is as follows:-

```
10 REM > Scr_convrt
20 REM Copyright : Gwyneth Pettit
   (Academic Software)
30 REM 3 April 1988
40 ON ERROR CLOSE#0:END
50 DIM buffer% 640, mask%(7)
60 MODE0
70 PROCset_up
80 PROCchange_dump
90 CLOSE#0
100 MODE mode%
110 REM Change colours by VDU19s
   here if required - for example,
120 IFmode%=4 ORmode%=0 VDU19,0,7,
   0;0,19,1,1,0;0
130 REM This changes white on black
   (default) to red on white
140 OSCLI("LOAD "+newfile$+addr$)
150 OSCLI("SCREENSAVE "+newfile$)
```

```
160 MODE mode%
170 OSCLI("SCREENLOAD "+newfile$)
180 END
185 :
190 DEFPROCset_up
200 !&80=149:!&84=-1
210 SYS 49,&80,&88
220 addr$=STR$(!&88)
230 PROCget_file
240 REPEAT
250   PRINT TAB(5,10);SPC55TAB(5,10)
     ;:INPUT"In which mode was it
     recorded (0,1,2,4 or 5 only) "
     ,mode%
260 UNTILmode%>=0 AND mode%<=5 AND
     mode%<>3
270 bytes_per_line%=20*2^(2-mode%
     DIV4)
280 no_of_bytes%=8*bytes_per_line%
290 length_of_screen%=32*no_of_bytes%
300 IF (mode%=0ORmode%=4) mask%(0)=0:
   mask%(1)=1:mask%(2)=2:mask%(3)=3
   :mask%(4)=4:mask%(5)=5 mask%(6)=6
   :mask%(7)=7
310 IF (mode%=1ORmode%=5) mask%(0)=4:
   mask%(1)=0:mask%(2)=5:mask%(3)=1
   :mask%(4)=6:mask%(5)=2:mask%(6)=7
   :mask%(7)=3
320 IFmode%=2 mask%(0)=6: mask%(1)=4
   :mask%(2)=2:mask%(3)=0:mask%(4)=7
   :mask%(5)=5:mask%(6)=3:mask%(7)=1
330 ENDPROC
335 :
340 DEFPROCget_file
350 REPEAT
360   REPEAT:CLS:PRINTTAB(8,2)"S C R
   E E N D U M P   C O N V E R T E R"
370   PRINTTAB(5,5);:INPUT"Give
     name of BBC screen dump
     to be converted ",file$
380   UNTILfile$<>" "
390   F%=OPENIN(file$)
400   UNTIL F%>0
410   newfile$=file$+"ARC "
420   PRINTTAB(5,7)"Your ARCHIMEDES
     screen is being saved as "newfile$
430   N%=OPENOUT(newfile$)
440   ENDPROC
445 :
450 DEFPROCchange_dump
460   PRINT TAB(5,15)STRING$(32,"-")
     TAB(5,15);
470   FOR S%=0 TO length_of_screen%-1
     STEP no_of_bytes%
480     start% = buffer%
490     PROCchange_lines
```

```

500 NEXT S%
510 ENDPROC
515 :
520 DEFPROCchange_lines
530 PRINT"+";
540 FOR B%=0 TO no_of_bytes%-1
550   ?(start%+B%)=BGET#F%
560 NEXT B%
570 FOR L%=0 TO 7
580   FOR B%=0 TO bytes_per_line%-1
590     old_byte%=?(start%+B%*8+L%)
600     PROCform_byte
610     IFmode%=2 nibble1%=byte%MOD16
        :nibble2%=byte%DIV16:BPUT#N%
        ,nibble1%*16+nibble1%:BPUT#N%
        ,nibble2%*16+nibble2%
620     IFmode%>3 pair1%=byte%DIV64:
        pair2%=(byte%MOD64)/DIV16:
        pair3%=(byte%MOD16)/DIV4:
        :pair4%=(byte%MOD4):
        BPUT#N%,pair3%*64+pair3%
        *16+pair4%*4+pair4%:
        BPUT#N%,pair1%*64+pair1%
        *16+pair2%*4+pair2%
630     IFmode%=0ORmode%=1 BPUT#N%
        ,byte%
640   NEXT B%
650 NEXT L%
660 ENDPROC
665 :
670 DEFPROCform_byte
680 byte%=0:mask%=&80
690 FOR D%=0 TO 7
700   byte%=byte%*2:IF(old_byte% AND
    mask%>>(7-mask%(D%)))>0 byte%+=1
710 NEXT D%
720 ENDPROC

```

Mode to Mode Converter

Mike Williams

This program will convert a screen from any Archimedes graphic mode to any other (though it has not been tested in mode 18 - 20).

Operation

The program asks the name of a screen file, the new mode and a name for the new screen file. The program uses different methods depending on the modes being converted. If both modes are 256 colour modes, then the screen is converted by copying screen memory byte by byte. If either the source file or the new mode has 256

colours, but not both, then:the array cmap%() is used to map between the colours.

The program is particularly good at converting 256 colour screens to 16 colour screens. (e.g. converting mode 15 screens, to mode 12 so that they can be dumped with the mode 12 colour printer)

Interesting "feature"

The *SCREENLOAD code doesn't seem to tell the VDU driver that the palette has been changed when a screen is loaded and relies on the fact that the VDU driver updates the hardware registers when the colours flash. If, however, flashing has been suppressed (e.g. by *FX 9,0) then the hardware registers don't get updated.

So, if you were wondering why SCREENLOAD sometimes doesn't change the colours to the colours that were SCREENSAVED, it's probably caused by an earlier program having left *FX 9,0 or *FX 10,0 set. *FX 9,25 and *FX 10,25 commands will restore things back to normal.

```

10 REM >ModeConv
20 *FX 9 25
30 *FX 10 25
40 DIM cmap%(255),bit%(8),
    data%(&34),d% 16
50 CLOSE #0
60 FOR i%=0 TO 255:cmap%(i%)=-1:
    NEXT
70 MODE12
80 REPEAT
90   INPUT"Source screen file:"file$
100  f%=OPENIN(file$)
110  IF f%<1 THEN PRINT"File not
    found"
120  UNTIL f%>0
130  FOR i%=0 TO &34
140    data%(i%)=BGET#f%
150  NEXT
160  oldmode%=data%(&34)
170  PTR#f%=12+data%(&30)+256*
    data%(&31)
180  PRINT file$ " is currently in
    mode ";oldmode%
190  CASE oldmode% OF
200    WHEN 0,4,18:bits%=1
210    WHEN 1,5,8,19:bits%=2
220    WHEN 2,9,12,20:bits%=4

```

Screen Mode Conversions

```

230 WHEN 10,13,15:bits%=8
240 OTHERWISE:PRINT "Source mode
      not catered for":END
250 ENDCASE
260 CASE oldmode% OF
270   WHEN 0,4,8,12,15,18,19,20:x%=2
280   WHEN 1,5,9,10,13:x%=4
290   WHEN 2:x%=8
300 ENDCASE
310 IF oldmode%>17 THEN y%=2 ELSE
      y%=4
320 bb%=-1
330 REPEAT
340   INPUT "New mode : "mode%
350   CASE mode% OF
360     WHEN 0,4,18:bb%=1
370     WHEN 1,5,8,19:bb%=2
380     WHEN 2,9,12,20:bb%=4
390     WHEN 10,13,15:bb%=8
400     OTHERWISE:PRINT"Destination
      mode not catered for"
410   ENDCASE
420 UNTIL bb%>0
430 INPUT "New file name : "newfile$
440 IF bb%=4 AND bits%<8 THEN *FX9
450 IF bits%=8 AND bb%<8 THEN
      mapdown%=TRUE ELSE
      mapdown%=FALSE
460 IF bits%<8 AND bb%=8 THEN
470   OSCLI("SCREENLOAD "+file$)
480   mapup%=TRUE
490   FOR i%=0 TO 15
500     SYS "OS_ReadPalette",i%,16
      TO ,,col%
510     r%=(col%>>8) AND &FF)>>5
520     g%=(col%>>16) AND &FF)>>5
530     b%=(col%>>24) AND &FF)>>5
540     t%=(r% AND 1)+(b% AND 1)+
      (g% AND 1)
550     cmap%(i%)=(r%>>1)+((g%>>1)
      <<2)+(b%>>1)<<4)+(t%<<6)
560   NEXT
570   ELSE
580     mapup%=FALSE
590   ENDIF
610 MODE mode%:OFF
620 IF bits%=8 AND bb%=8:bytecopy%
      =TRUE ELSE bytecopy%=FALSE
630 nextmap%=0
640 mask%=(1<<(bits%)-1)
650 Z%=&FF
660 IF bytecopy% THEN PROCbytecopy
      ELSE PROCconvert
670 OSCLI("SCREENSAVE "+newfile$)
680 END

700 DEFPROCconvert
710 IF oldmode%=2 THEN bits%=8
720 byte%=BGET#f%
730 FOR Y%=1023 TO 0 STEP -y%
740   FOR X%=0 TO 1279 STEP x%
750     C%=(byte%) AND mask%
760     byte%=byte%>>bits%
770     Z%=Z%>>bits%
780     IF Z%=0 THEN
790       Z%=&FF
800       byte%=BGET#f%
810     ENDIF
820     IF mapdown% OR mapup% THEN
830       IF cmap%(C%)=-1 THEN
      PROCmap
840       GCOL cmap%(C%)AND&3F TINT
      cmap%(C%)
      ELSE
850       GCOL C%AND&3F TINT C%
860     ENDIF
870   RECTANGLEFILL X%,Y%-4,8,4
880   NEXT
890 NEXT
900 NEXT
910 ENDPROC
930 DEFPROCmap
940 FOR i%=0 TO 7
950   IF (C% AND 1<<i%) THEN bit%
      (i%)=1 ELSE bit%(i%)=0
960 NEXT
970 r%=(2*bit%(4)+bit%(2))*64
980 g%=(2*bit%(6)+bit%(5))*64
990 bl%=(2*bit%(7)+bit%(3))*64
1000 t%=(2*bit%(1)+bit%(0))*16
1010 IF nextmap%<16 THEN COLOUR
      nextmap%,(r%+t%),(g%+t%),(bl%+t%)
1020 cmap%(C%)=nextmap%
1030 nextmap%+=1
1040 ENDPROC
1060 DEFPROCbytecopy
1070 d%!0=149
1080 d%!4=-1
1090 SYS "OS_ReadVduVariables",d%
      ,d%+8
1100 scrnaddr%=d%!8
1110 IF mode%=15 THEN N%=163839:S%=2
      ELSE N%=81919:S%=1
1120 FOR p%=0 TO N% STEP S%
1130   byte%=BGET#f%
1140   scrnaddr%?p%=byte%
1150   IF S%=2 THEN scrnaddr%?(p%+1)
      =byte%
1160   IF oldmode%=15 THEN junk%=
      BGET#f%
1170 NEXT
1180 ENDPROC

```

Watford Video Digitiser Review

Matthew Treagus

The Watford Electronic Real-Time Video Digitiser is a podule (or 'expansion card' as Acorn now prefer to call them) supported with a disc containing utilities, demos and examples. The main podule software is supplied on ROM and is contained on the podule itself. Both the software **and** the hardware have been designed by Mike Harrison so, as you would expect, the two fit together.

The software is extremely professional, allowing the very best to be gained out of the digitiser. The Digitiser Module (written to conform to Acorn protocols) is accessed via either *commands or SWI calls allowing you to capture your own digitised pictures quickly and easily. The simplest way to grab an image is to use a 'key triggered grab'. A 'hot key' is employed that will grab an image when pressed. The key used to do this can be configured and the effect of the key can also be configured.

The *command side of things provides a nice top end to the digitiser and some very good effects can be achieved using the many options that can be set following the command. For more advanced use, the SWI's can produce complex results under very flexible parameters. The software has obviously been well planned and thought through because everything is catered for, the software fully supports multisync monitors and these can be used to produce very high definition images. Many monochrome monitor owners will have found their monitors are not too useful when in 256-colour modes but now they can be used to produce the full 64 shades of grey (or green) that are not available on colour monitors. (It is not good enough to just switch your colour monitor to 'GREEN' because this is not the same as a monochrome. All 'GREEN' does on your colour monitor is turn off the red and blue guns and only use the green one.)

ROM Software

Let me try to explain briefly the facilities contained in the module:-

Filing - The module contains six different filing commands that operate on three different file types (a load and save for each type)

There is firstly a *PicSave and *PicLoad that allow the 128k of video RAM on the main board to be saved and loaded. The Picture files are stored in a compacted form to save disc space.

A *FastSave and *FastLoad have also been provided to improve on Arthur's painfully slow *Screenload and *Screensave. They are similar to the module published in Archive Issue 6 although this version takes just 5 seconds to save a MODE12 screen. In addition to these there is a *GrabSave that can be used to grab frames from the digitiser and store them on disc at the touch of a button.

The third set of commands allows "Pixel Look-up Tables" to be filed. These allow you to change the apparent value of certain shades in the image. These can be used to obtain "contrast bending", where extreme shades are emphasised, or to obtain simple effects such as negatives or thresholding (colours above a certain shade become white and those below become black). The creation and editing of these files is not quite as easy as I would have hoped, but several files were supplied on the disc and they provided all I needed.

Grabbing and displaying - The display of a digitised image requires two actions; Grabbing the incoming image and then transferring it to the screen, many of the display routines do both.

Before an image can be displayed, the palette must be set to the grey levels. All grab routines will do this unless the machine configuration or the command options specify otherwise. The palette can also be set using *Bwpal. This command can also be used to produce different

amounts of greys. e.g. *Bwpal 2 produces just black and white.

The actual grabbing of an image is done by the command *Grab which pulls an image into the video RAM and displays it. You can turn your monitor into a rather effective black and white television using *See. The digitiser and software can display the grabbed screens at up to 25/second if the quarter screen option is set. This command streams images from the digitiser to the screen until a key or mouse button is pressed.

I hope by now I have made clear the difference between the digitiser video RAM on the digitiser and the screen RAM on the actual Archimedes. To see the picture stored in the video RAM *ShowPic is used. In conjunction with this command and "key triggered grabbing" the *Section command can be used to specify a section of the screen which will be blown to full size when displayed. Another useful command *MakeSprite allows you to make a sprite file from all or part of the current digitised picture.

Once an image has been captured it can be either scaled or rotated by SWI commands which work amazingly quickly. Although the resolution of the digitiser is good and the analogue to digital conversion clean, there may still be blips and fuzzes on the screen caused by bad cables or bad video signals, these can be removed using two different commands: *Zit removes unwanted "noise" from the picture and *Smooth averages out each pixel with its neighbours.

The software contains a very high quality printer dump routine and even my GLPII with a dry ribbon produced some very respectable output. The dumps work with all Epson compatibles or printers supporting ESC"" graphics. The *PicDump command is a very versatile screendump facility and the output was very good, though I have tried the dump routines on a wide load carriage. The dump routine supports images up to 19 inches square and can print in eight different orientations. The dumps use printer dot patterns, seven of which are built into

the module, and a printer pattern editor is supplied on the disc. The pattern files are directly compatible with the Artisan "PrintPat" files. Unusual printer types are supported by the "Printer information block" that allow codes to be defined for each different printer. A program is supplied on disc to allow the use of Epson MX compatible printers.

All of the above commands can have various strings of options attached to them such as, use current mode, do not set palette, plot only between last two graphics points or mouse clicks, use pixel lookup table, use quarter screen only and don't grab but use current image. These options have changed what would have been simple instructions into powerful commands.

The Hardware

The hardware takes a standard 1v composite video signal and the digitiser will adjust its input range over the approximate range 0.5 to 1.5 volts. A composite video signal can be found on all videos and on some TV Tuners. The digitiser also has a terminating link that can be removed if terminating resistance is not required.

Extra programs

The disc supplied with the digitiser comes complete with loads of clever effects including a 'colour grabber' that works by using a video camera with coloured filters. The disc also contains several other images and lookup tables.

Documentation

The glossy manual is clearly written and, as far as I can see, error-free. It provides a complete documentation of programs on the utility disc and the digitiser software and hardware. The manual clearly explains ways of integrating the trigger key grabbing and digitised images into any existing art package, including how to incorporate *FastLoad and *FastSave into Artisan and Arctist.

Conclusion

I was very impressed with this package, finding no faults or things that could have been

Operating System Commands from BASIC

Ian Smith

BBC Basic V on the Archimedes is acknowledged to be both fast and comprehensive. There are times, however, when a little more speed or access to some operating system routines is required. On many systems this means resorting to assembly language but the Archimedes provides one facility which makes more speed and low level access possible.

The SYS command gives the programmer the ability to place values into the registers of the Risc chip, call the OS interrupt routine and retrieve the results from the registers. The operating system provides SWI's (SoftWare Interrupts) which range from those familiar to BBC Micro owners, OS_Byte and OS_Word, to suites of calls to the Wimps, Fonts, Sprites, Sound etc. To use SYS is straightforward and some examples will show the syntax and usage of this versatile and powerful command. However, the Programmers' Reference Manual will be needed for the register requirements of any other calls.

Simple Example

Each SWI has an associated number and, in many cases but not all, a name. An example is SWI &45 named "OS_Plot" which is equivalent to the BASIC PLOT command (VDU 25,...). This SWI requires 3 values to be stored in the registers R0, R1 and R2 representing the Plot command number, x-coordinate and y-coordinate respectively. It is much quicker than using VDU 25 and just as easy.

improved. I was very pleased to see all the features well supported by both *commands and SWI calls. The trigger key grabbing makes the use of the digitiser much easier in Artisan and other art packages. I would have liked a command like "*CompactSave" that saved standard screens in a compact form (but perhaps there is no room left in the ROM!!). A "pixel lookup table" editor would also have been nice.

VDU 25,69,640;512; (Plot a point at 640, 512 in current foreground colour) becomes :

```
SYS &45,69,640,512  
or SYS "OS_Plot",69,640,512
```

This is an example of a SYS call which does not return a result and its general form is :

```
SYS &NN ,R0value,R1value,  
R2value,R3value etc
```

where NN is SWI number and R0value etc are the values to be placed in the registers. If any register, R0 for example, does NOT require a value but others, R1 and R2, do then use :

```
SYS &NN ,,R1value,R2value  
R3value etc
```

where the ,, indicates that no value is being passed to R0.

Returning parameter values

Many SWI calls produce results which are stored in registers and which need to be extracted, e.g. "OS_Mouse" (&1C), the call to read the current mouse values (x-coordinate, y-coordinate and button pressed). This requires no values to be placed in the registers but returns 4 values via registers R0-R3. The first 3 are the same as those returned by the BASIC command MOUSE X,Y,B and the fourth is the time that the last mouse entry was buffered. It can be used as follows:

```
SYS "OS_Mouse" TO X,Y,B,T
```

The 'TO' indicates that the values in the registers R0, R1, R2 etc are to be stored in the

Over all though the package contains very good hardware and extremely well presented software to back it up. The only drawback is the price! You have to decide if it is really worth £250 (+VAT). Well done Mike Harrison! Well done Watford Electronics! **A**

£288 inclusive of VAT & U.K. carriage
when ordered through Archive Magazine.
Plus free demo disc!

variables X,Y,B,T etc respectively.

In General: SYS &NN TO ValueOfR0, ValueOfR1, ValueOfR2 etc

If any registers do not contain values for returning then they are omitted as in the following where only R1 contains a value :

```
SYS &NN TO ,ValueOfR1
```

N.B. The comma is essential to indicate that the value in R0 is not required.

In general, calls will both pass values and return results. An example is a call to read the Font Manager version number and usage of the Font Cache "OS_Font_CacheAddress" (&40080). This requires R0 to contain 0 on entry and on return R0 contains the version number, R1 the amount of Font Cache used and R2 the total amount of font cache available.

```
SYS &40080,0 TO Version,Used
                                ,Total
```

will call this SWI and store the results in Version,Used,Total.

In general a SYS call is :

```
SYS &NN ,R0val,R1val,R2val,
    R3val TO VarR0Val,VarR1Val
                                ,VarR3Val
```

The use of "OS_Byte" requires that the particular call code is placed in R0 and any results are returned via the number of registers required. An example is the call equivalent to INKEY(N) where N is >=0 (OS_Byte &81).

This requires &81 to be placed in R0 and for R1 and R2 contain the LSB and MSB of the value of N. The result (the ASCII code of the key pressed, if any) is returned via R1. The equivalent call to Num = INKEY(N) is therefore :

```
SYS "OS_Byte",&81,(N MOD 256),
    (N DIV 256) TO ,Num
```

But Num = INKEY(100) can be coded as SYS "OS_Byte",&81,100 TO ,Num

As some of the more recent and less used SWI calls have equivalent names it may be safer to use the number for these. There is even a SWI to

convert a SWI number into its equivalent name!

Simple Example Programs

```
10 REM > SYSeg
20 REM Ian Smith
30 REM A Few examples
40
50 MODE 12
60
70 REM A Circle Using OS_Plot
80 GCOL 0,2
90 SYS "OS_Plot",&44,1000,512 :
    REM Move to centre
100 SYS "OS_Plot",&9D,1000,612 :
    REM Draw Circle
110 SYS "OS_Plot",&44,640,512 :
    REM Move to Centre of screen
120
130 REM Repeat 'til space bar pressed
140 REM Use GET equivalent
150 PRINT "Press Space Bar"
160 REPEAT
170     SYS "OS_ReadC" TO AsciiCode
180 UNTIL AsciiCode =32
190
200 REM Check Font Details
210 SYS&40080,0 TO Version,Used,Total
220 PRINT " Font Version ";Version
    /100
230 PRINT " Cache Used ";Used DIV
    1024;"K"
240 PRINT " Available ";Total DIV
    1024;"K"
250
260 REM Now the Mouse
270 PRINT "Use Mouse to draw. Press
    ALL 3 Buttons to stop"
280 REPEAT
290     SYS "OS_Mouse" TO X,Y,B,T
300     REM We don't need T
310     DRAW X,Y
320 UNTIL B=7
330
340 PRINT "Press RETURN"
350 REM Use INKEY(100) equivalent
360 REPEAT
370     SYS "OS_Byte",&81,100 TO ,Num
380 UNTIL Num=13

10 REM > SWIList
20 REM Ian Smith
30 REM Simple SWI Lister Using SYS
50 N=40
60 DIM Buffer% N
70
```

IEEE488 Podule

Brian Cowan

The IEEE488 interface bus, otherwise known as the General Purpose Interface Bus (GPIB), or Hewlett Packard Interface Bus (HPIB), is an eight bit parallel bus which includes handshake, control and address lines to access up to fifteen individual instruments. If one is attaching only commercial instruments to the computer and if ultra high speed data transfer is not required, and if all ones instruments have an IEEE488 bus interface, then all communication can be performed down this bus. Thus, for many interfacing requirements, an IEEE488 interface is all that is required. In practice, however, some instruments will be interfaced in this way and some by other means.

Hardware

The IEEE488 interface Podule for the Archimedes is manufactured by Intelligent Interfaces – the same people who made the IEEE488 interface for the earlier BBC machines. It provides a full implementation of the IEEE standard 488-1978 including supplement IEEE standard 488A-1908. The heart of the interface is the bus controller chip, Texas Instruments TMS9914A, again, the same as in the previous interface. This chip is an intelligent controller specifically designed for IEEE488 applications. It relieves the processor of the task of handling IEEE488 protocol when commands and data are sent and received and implements most of the IEEE488 functions. If processor intervention is required in order to support a function, the bus controller chip generates an interrupt. A ROM on the Podule

board contains support code that loads into a relocatable module. The address of the computer may be set by a DIP switch on the board and the computer may be configured either as a System Controller or as a Simple Talker Listener

Software Implementation

The original BBC IEEE488 interface was implemented as a filing system. Reading and writing to instruments was performed from BASIC using INPUT# and PRINT# statements following calling the filing system. This was somewhat inconvenient. Furthermore the filing system software used interrupts for everything, which was rather slow. A later version of the software changed all this. A collection of BASIC procedures and functions was provided to perform the communication commands, and the system no longer made extensive use of interrupts. This resulted in much faster and more convenient operation. The Podule for the Archimedes operates in the same manner. A BASIC library of functions and procedures is provided on the software distribution disc. This no longer needs to be appended to the current program as the LIBRARY and INSTALL statements of BASIC V may be used. Much software may therefore be transferred directly from previous BBC machines. Problems that arise are likely to be due to the incredible speed of the Archimedes – the Podule operates up to a staggering 250 kbytes per second! The comprehensive manual includes a section on using the IEEE488 interface from assembler.

Conclusions

An IEEE488 interface is vital for controlling many scientific instruments. This Podule from Intelligent Interfaces is fast and particularly convenient to operate from BASIC. Support for other languages should be forthcoming. **A**

```
80 REM TRY ALSO FOR Code = &40000
                                TO &41000
90
100 FOR Code = &0 TO &200
110   $Buffer% = STRING$(N, " ")
120   SYS &38, Code, Buffer%, 255
130   PRINT ~Code; " "; $Buffer%
140 NEXT Code A
```

More Help on Using View

Brian Cowan

Being a stick-in-the-mud who is more than happy with View for creating short non-scientific documents, I was delighted that I could get it to run on the Archimedes with 65Arthur, the 6502 emulator. The arrival of the bug-ridden ArcWriter did nothing to change my mind. However, in getting this classic (pre-historic?) wordprocessor to work on the Archimedes I encountered a number of problems and quirks. In solving the problems and understanding the quirks I learned a few things about the operation of View and the implementation of the emulator. I strongly recommend serious users of View to read Bruce Smith's View Dabhand Guide. This is a superb book.

Which View?

The ROM versions of View 3 will not run from RAM without judicious poking of bytes as explained in Archive No.3 page 10. It is the RAM version that comes with the Master Compact, View B3.11, that I used. This is one of the later versions and has a number of extra features: an Epson printer driver is included and top-bit set "extended ASCII" characters are supported. The keystrip for the function keys is the same for all versions of View. A suitable keystrip was given away free with issue 2 of Archive and extra ones may be purchased for minimal(?) cost. If you are going to make your own then remember that the f0 key corresponds to the PRINT key on the Archimedes. This could be redefined but I am happy to leave it there.

Creating a !BOOT file

My plan of action was to create a View disc that could be booted up by the usual <shift-break> procedure. The root directory contained the boot file !BOOT together with two directories. The first directory was called TEXT, in which all View text files would reside, and the second was called Library and this contained the View ROM image, 65Arthur and a startup View file called BLANK.

The !BOOT file contained the following:

```
QUIT
*FX 202,16
RMRUN LIBRARY.65ARTHUR
```

```
*GO F800
LOAD LIBRARY.VIEW 8000
GO 8000
SETUP FJI
*DIR TEXT
LOAD $.LIBRARY.BLANK
```

The Quit command leaves ARM BASIC in readiness for loading the 65Arthur relocatable module. The *FX command on the next line has the effect of turning off the caps lock key. Thus View is entered with the keys giving lower case letters as on a typewriter. The third line loads and runs the 6502 emulator and the machine now comes up in the emulator's BASIC 4. The next line will be familiar to Turbo and 6502 second processor users. Remember that 65Arthur is emulating a second processor system. In a second processor the Tube Operating System starts at &F800. Thus *GOing to that location puts the machine into the "no language" environment. Next the View image is loaded into the "ROM" space between &8000 and &C000. These addresses are the emulated addresses, not true Archimedes locations, which are &10300 and &14300. When View is loaded here it overwrites the pre-existing (emulator) BASIC 4. The machine is then sent back to &8000 whereupon View is entered.

The SETUP command applies only in View 3. This sets up the text screen in Format, Justify and Insert modes. This is how I like it. Others may prefer a different arrangement. The initial file called BLANK is then loaded into View from the LIBRARY directory. This file contains my preferred ruler together with settings for the margins and the page length. This saves me typing the same thing in at the start of each new document. It is important that this file has ACCESS attributes LR so that it can not be overwritten inadvertently. The disc must be set to *EXEC the !BOOT file by typing *OPT 4 3.

"Top-bit set" characters

The ASCII standard is a seven bit code. The 128 characters in the specification include all the usual symbols on the keyboard, both upper and lower

Operating Systems Galore

Brian Cowan

As many people realise, the current version of Arthur, 1.2, is beset with bugs. Some of these have been covered in these columns and Acorn have produced a three page list! An updated Arthur should be available at some stage. The Serial Port problem is slightly different since there are problems with the 65C51 serial chip and chips from different manufacturers have different problems. The latest (final?) version of the Serial Port patch Relocatable Module, version 1.24, is claimed to solve the hardware faults together with the bugs that crept in (crawled in?) to the Serial Port software in Arthur 1.2. Ironically, Rockwell are now producing a bug-free 65C51 but of course that will not work in the Archimedes!

Arthur 2.0 and ARX

As well as ARX, the Unix-like operating system that Acorn are working on, there is a new version of Arthur, number 2.0. It is rumoured that Arthur 2.0 will support a sort of multi-tasking together with numerous other enhancements, while still being clearly recognisable as a BBC operating system. There is not much information on ARX. It is supposed to be user-friendly (unlike true Unix) and it will sit on top of Arthur. That is, it will be loaded into RAM and make calls to Arthur as necessary. The alternative strategy would be to replace the Arthur ROMs with new ones thus producing a machine with a completely new "personality".

case. In practice each character is represented by a single byte. Thus for the ASCII standard characters the top bit of the byte is zero or "unset". It makes sense to extend the character set to 256, the new characters having a one for their top bit; the top bit is "set". The problem is that there is no universal standard for the new characters. In fact the Archimedes includes four Latin and a Greek font together with those of the old BBC machines. The character set in use is determined by *Configure Alphabet or *Configure Country.

Impulse from Computer Concepts

Acorn had better watch out however, since Computer Concepts are planning to launch their own operating system known as Impulse for the Archimedes. This is supposed to support sophisticated multitasking with full memory protection: one task can crash while others continue unaffected. This system is designed to replace Arthur so it is not clear exactly what facilities will be provided. Acorn have always acknowledged that making a BBC-like operating system for the Archimedes has limited the power of RISC technology but they are producing a BBC machine and continuity is vital.

Clearly, Computer Concepts are not restricted by such considerations in producing Impulse. It appears that the motivation for developing Impulse was dissatisfaction with the facilities of Arthur, particularly as an operating system to support desktop publishing applications. We can anticipate a number of software packages from Computer Concepts that run under Impulse. Running Impulse from RAM on a 310 machine would not leave much RAM space for the user. Thus it will probably be supplied as a ROM chip set with a header to plug in together with Arthur, having a software switch. It remains to be seen how useful Impulse will be for other applications. **A**

The top-bit set characters are obtained with the Alt keys which sit either side of the space bar. If you press the <ctrl-shift-alt> and then release them, when the next key is pressed 128 will be added to its code in the keyboard buffer. The top-bit set character of the appropriate font will appear on the screen. The earlier versions of View ignored the top bit of character bytes, but these are accepted by Master Compact versions such as B3.11. Thus using this version, a full extended character set may be used within the word processor. Printing the extra characters is not simple; this will be discussed in a future article. **A**

Colour Printouts – The Easy Way

W.R.Davis

If you cannot afford a colour printer to preserve your graphics masterpieces why not photograph them? The main difficulty is the determination of the correct exposure but if the following hints are observed good results can be obtained.

If possible, use a 35mm single-lens reflex fitted with a zoom lens – either 35–85mm or 80–210mm. You can use a standard 50mm lens but it should be capable of focussing down to about 14 to 15 inches. A tripod is virtually essential but passable results are obtainable with the camera propped up or perched on books, blocks etc. As a refinement, if the camera is fitted with Delayed Action use this to eliminate camera shake. 100 ASA print or 64 ASA slide film is recommended for optimum quality – but you can use faster film if you wish. If the camera does not have a built-in meter, a hand-held exposure meter can be employed but do not bring it too close to the monitor screen as there is a risk that the electromagnetic fields produced by the monitor scan coils may interfere with the meter's moving coil system.

Set up a typical display on the monitor and adjust brightness and contrast for a well graded picture. Once this has been done these controls **must not** be touched again or the whole calibration procedure will have to be repeated.

Align the camera, pointed accurately at right angles to the centre of the screen and either 'zoom' the lens or adjust the camera distance so that the screen image, as seen through the eyepiece, slightly overlaps the frame. Set the camera aperture to f8 and focus as accurately as possible using the screen 'spots' or 'stripes'. The camera position should remain fixed from now on. Type in VDU19,0,7,0,0,0 to give an all-

white screen, switch off the room lights and note the meter reading. (If you are working in daylight, draw the curtains!)

This reading is the 'peak white' exposure value and the correct exposure is obtained by multiplying this by 4. This can be done by either 'opening up' the lens by two stops e.g. from f8 to f4 or increasing the exposure time fourfold e.g. from 1/4s to 1s. This 'corrected' exposure should be used for all future photographs. Before starting a new photo session, a check could be made with the 'peak white' screen with adjustment of the brightness to give the standard 'peak white' value.

The camera should be on MANUAL mode but if only AUTO is available note the correct exposure and, when the desired image has been restored to the screen, adjust the ASA film speed control to give this reading. Nowadays many cameras have an exposure compensating control with a range of +2 to -2 stops and this could be used as an alternative. If all else fails assume an exposure of 1s at f8 and do a series of test exposures 'bracketting' this value e.g. from 4x (1s at f4) to 0.25x (0.25s at f8) and take the best as your standard exposure. Remember that opening up from f8 to f5.6 (or from f5.6 to f4) is equivalent to doubling the exposure time. The room lights need only be off for the calibration and the actual exposure. The latitude of modern colour films is sufficient to give acceptable results with moderate over or under exposure but it is advisable to keep a record of the exposure made for a particular image so that a certain amount of 'fine tuning' can then be carried out later. **A**

Fact-File

ABACUS Training	29 Okus Grove, Upper Stratton, Swindon, Wilts, SN2 6QA.
Academic Software	Sourby Old Farm, Timble, Otley, Yorks, LS21 2PW. (0943-88-628)
ACE Computing	27 Victoria Road, Cambridge, CB4 3BW. (0223-322559)
Blue Grey Software	28 Bobbitts Way, Wivenhoe, Essex, CO7 9NJ.
Brainsoft	22 Baker Street, London, W1M 1DF. (01-486-0321)
CCD Computer Services	71 Marlborough Park Avenue, Sidcup, Kent, DA15 9DL. (01-302-5427)
CJE Micros	78 Brighton Road, Worthing, W Sussex, BN11 2EN. (0903-213361)
Clares Micro Supplies	98 Middlewich Road, Rudheath, Northwich, Cheshire, CW9 7DA. (0606-48511)
Colton Software	149-151 St Neots Road, Hardwick, Cambridge, CB3 7QJ. (0954-211472)
Computer Concepts	Gaddesden Place, Hemel Hempstead, Herts, HP2 6EX. (0442-63933)
Computer Spares Ltd	61A Braemar Drive, Highcliffe, Dorset.
Contex Computing	15 Woodlands Close, Cople, Bedford, MK44 3UE. (02303-347)
Dudley Micro Services	30 Hadley Close, Netherton, Dudley, DY2 9JX. (0384-633142)
EMR Ltd	14 Mount Close, Wickford, Essex, SS11 8HG. (0702-335747)
Fairhurst Instruments	Dean Court, Woodford Road, Wilmslow, SK9 2LT. (0625-525-694)
GEM Electronics	17 Tandragee Road, Portadown, Craigavon, BT62 3BQ.
HopeSoft	Hope Cottage, Winterbourne, Newbury, Berks, RG16 8BB. (0635-248472)
Intelligent Interfaces Ltd	43b Wood Street, Stratford-upon-Avon, CV37 6JQ. (0789-415875)
Jansons	93 North Street, Stilton, Peterborough, PE7 3RR.
Lingenuity	P.O.Box 10, Halesworth, Suffolk, IP19 0DX. (0986-85-476)
McSoft	36 Alfred Street, Dunstable, Beds. (0582-699-483)
Magenta Research Ltd	Amp House, Dingwall Road, Croydon, CR0 9XA. (01-681-7179)
Minerva Systems	69 Sidwell Street, Exeter, EX4 6PH. (0392-37756)
Mitre Software	26 Creechurch Lane, London, EC3A 5BA. (01-283-4646)
Pineapple Software	39 Brownlea Gardens, Seven Kings, Ilford, Essex, IG3 9NL. (01-599-1476)
RESOURCE	Exeter Road, Doncaster, DN2 4PY. (0302-63800/63784)
Silicon Vision Ltd	Signal House, Lyon Road, Harrow, Middlesex, HA1 2AG. (01-422-2274)
Texellence	2 Greenhill Road, Coleraine, N Ireland, BT51 3JE.
Norwich Computer Services	18 Mile End Road, Norwich, NR4 7QY. (0603-507057)

Archive

Subscription Magazine and Support Group for *Archimedes* users

Archive Magazine contains:

- News
- Reviews
- Hints and Tips – a major feature
- Articles for Beginners
- The Latest Technical Information
- Program Listings
- Free Small Ad's Section
- HELP – Requested and Offered
- Contact Box – to help you form common interest groups

Eureka! – Bulletin Board

0603-250689 on 1200/75 or 300/300

- Very Latest News
- Down-load Software
- Mailbox Facilities
- Chat line

Technical Help Service (£8 / year)

A telephone hot-line service for immediate help with your technical problems. Any member can send written enquiries, but for a fast response use the THS!

Members' Discount: 7.5% off software from Computer Concepts, Minerva Systems and Clares Micros Supplies purchased through Norwich Computer Services.

Subscription: 12 issues £12.50 (UK)
Europe £18, Middle East £22,
America / Africa £25, Elsewhere £27.
Technical Help Service £8

N.B. All earlier issues have now been re-printed – you may back-date your subscription as far as issue 1 (October 1987) – to take advantage of this huge bank of information.

Archimedes is a trademark of Acorn Computers Ltd.

* Please send copies of *Archive* magazine for one year starting from
Volume 1 Issue _____

* Please enrol me on the Technical Help Service for one year. (£8)

I enclose a cheque for £ _____ payable to "Norwich Computer Services".

Name: _____

Address: _____

Postcode: _____

Norwich Computer Services, 18 Mile End Road, Norwich, NR4 7QY